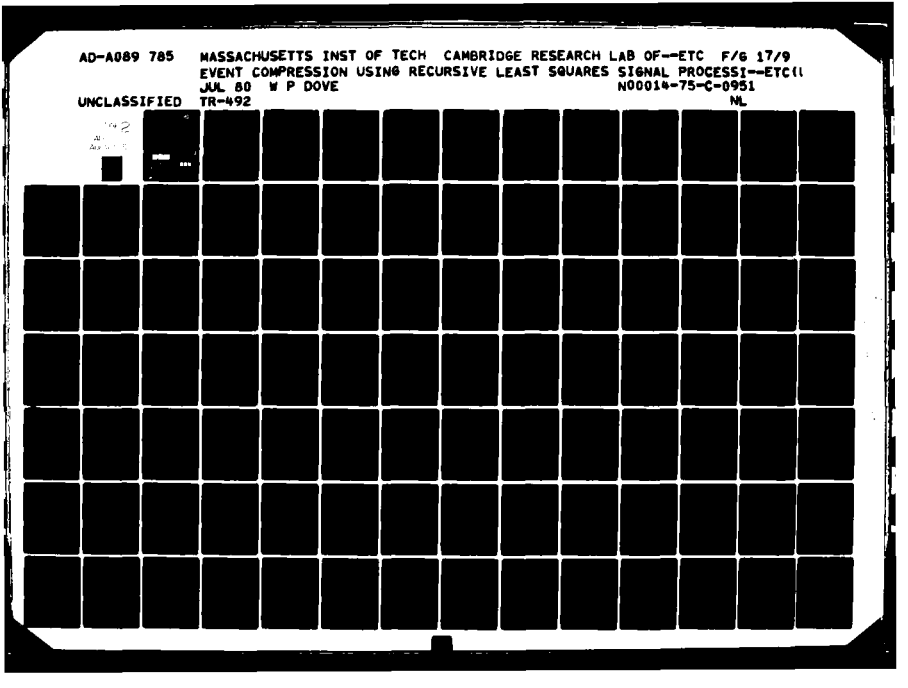
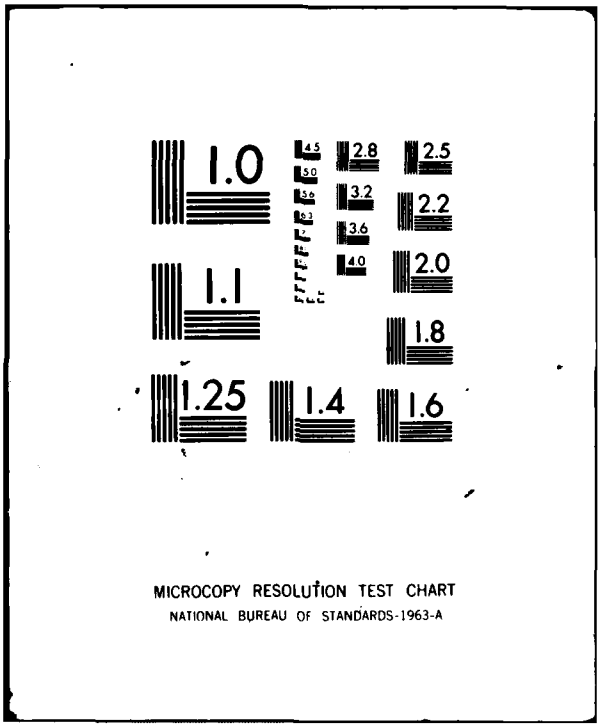
AD-A089 785 MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF--ETC F/G 17/9
EVENT COMPRESSION USING RECURSIVE LEAST SQUARES SIGNAL PROCESSI--ETC(U
JUL 80 W P DOVE N00014-75-C-0951
UNCLASSIFIED TR-492 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

"Event Compression Using Recursive Least Squares Signal Processing"

Webster Pope Dove

TECHNICAL REPORT 492

July 1980

Massachusetts Institute of Technology
Research Laboratory of Electronics
Cambridge, Massachusetts 02139

This work was supported in part by the Advanced Research Projects Agency monitored by ONR under Contract N00014-75-C-0951-NR 049-328 and in part by the National Science Foundation under Grants ENG76-24117 and ECS79-15226.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A089 785	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) EVENT COMPRESSION USING RECURSIVE LEAST SQUARES SIGNAL PROCESSING.		5. TYPE OF REPORT & PERIOD COVERED (7) Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Webster P. Dove		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-75-C-0951 NSF-ENG76-24117
9. PERFORMING ORGANIZATION NAME AND ADDRESS Research Laboratory of Electronics Massachusetts Institute of Technology Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 049-328
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22217		(11) 12. REPORT DATE July 1980 13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	(12) 154	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) (9) Master's thesis Approved for public release; distribution unlimited (14) TR-492		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) recursive least squares event compression linear prediction		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) see other side		

DD FORM 1 JAN 73 1473 304050

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract

This work presents a technique for time compressing the events in a multiple event signal using a recursive least squares adaptive linear prediction algorithm. Two event compressed signals are extracted from the update equations for the predictor; one based on the prediction error and the other on the changes in the prediction coefficients as the data is processed.

Using synthetic data containing three all-pole events, experiments are performed to illustrate the performance of the two signals derived from the prediction algorithm. These experiments examine the effects of initialization, white gaussian noise, interevent interference, filtering and decimation on the compressed events contained in the two signals.

EVENT COMPRESSION USING RECURSIVE LEAST SQUARES SIGNAL PROCESSING

by

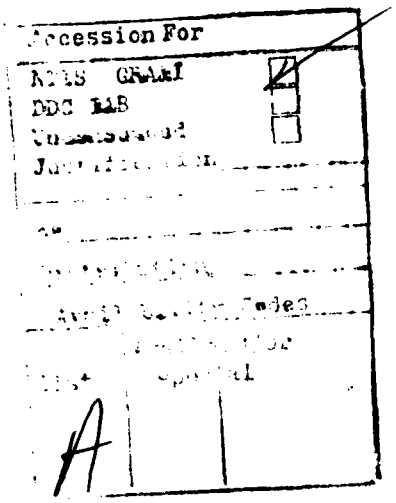
Webster Pope Dove

Submitted to the Department of Electrical Engineering and Computer Science, on 15 July 1980, in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering.

ABSTRACT

This work presents a technique for time compressing the events in a multiple event signal using a recursive least squares adaptive linear prediction algorithm. Two event compressed signals are extracted from the update equations for the predictor; one based on the prediction error and the other on the changes in the prediction coefficients as the data is processed.

Using synthetic data containing three all-pole events, experiments are performed to illustrate the performance of the two signals derived from the prediction algorithm. These experiments examine the effects of initialization, white gaussian noise, interevent interference, filtering and decimation on the compressed events contained in the two signals.



THESIS SUPERVISOR: Alan V. Oppenheim

TITLE: Professor of Electrical Engineering

To Monica

Acknowledgements

I give my thanks to all the people in the Digital Signal Processing Group at MIT. Many fruitful discussions with them at odd hours contributed to this work. Of particular help were Dr. Lloyd Griffiths with his insight into least squares algorithms and Steve Lang for infinite patience in listening to my ramblings. Last and most I thank Prof. Alan Oppenheim for the original problem suggestion, much needed guidance during the investigation and the pressure necessary to produce this document. If I can acquire a tenth of his clarity of thought while I am here, I will consider myself lucky.

CONTENTS

1. INTRODUCTION

In various signal processing problems, signals arise which contain pulses or events that must be located or detected. Problems of this type include RADAR and SONAR rangefinding, speech pitch detection and seismic data analysis. All of these fields share the need for a signal processing technique which can compress events occuring in the raw data into shorter events in the processed data. This type of processing can reduce the overlap between successive events leading to improved detectability and by increasing the impulsiveness of each event make locating the events easier. Such a signal processing procedure, which reduces the duration of events in an input sequence without changing their relative separations, is an event compression algorithm. Examples include homomorphic deconvolution, matched filtering, linear predictive deconvolution.

Let us illustrate the application of event compression. Consider the problem of finding the range of multiple targets with RADAR or SONAR. To reduce peak power requirements in transmission, the source pulses are dispersed in time. In addition, the reflection functions of the targets may cause the returns to be spread out even more. Therefore the events in the received signal must be compressed if the targets are to be located accurately.

Surface seismograms are created by generating a disturbance at the surface of the earth, either with a mechanical vibrator or with an explosion, and recording the subsequent seismic vibrations with an array of geophones also located at the surface. As waves propagate into the earth, they are partially reflected at boundaries between differing layers and ideally the recordings made at the surface can be used to locate the depths of these boundaries and their reflectivities. As in the RADAR situation, the duration of the source pulse and the extension of that duration by the individual reflection functions leads to the need for event compression of the data.

One approach to vocal pitch estimation is to measure the time between successive glottal pulses. However, since these pulses are filtered by the vocal tract impulse response before emanating from the lips, the speech waveform does not offer well defined points at each cycle from which to measure the period to the next cycle. It is therefore necessary to compress the pulses of the speech waveform without changing their positions for this scheme to be effective.

Another example is the problem of sonic well logging which motivated this thesis. The procedure for generating a sonic well log is to lower a tool, shown schematically in figure 1.1, down an oil well and then raise it at a fixed rate

Well Log
Diagram

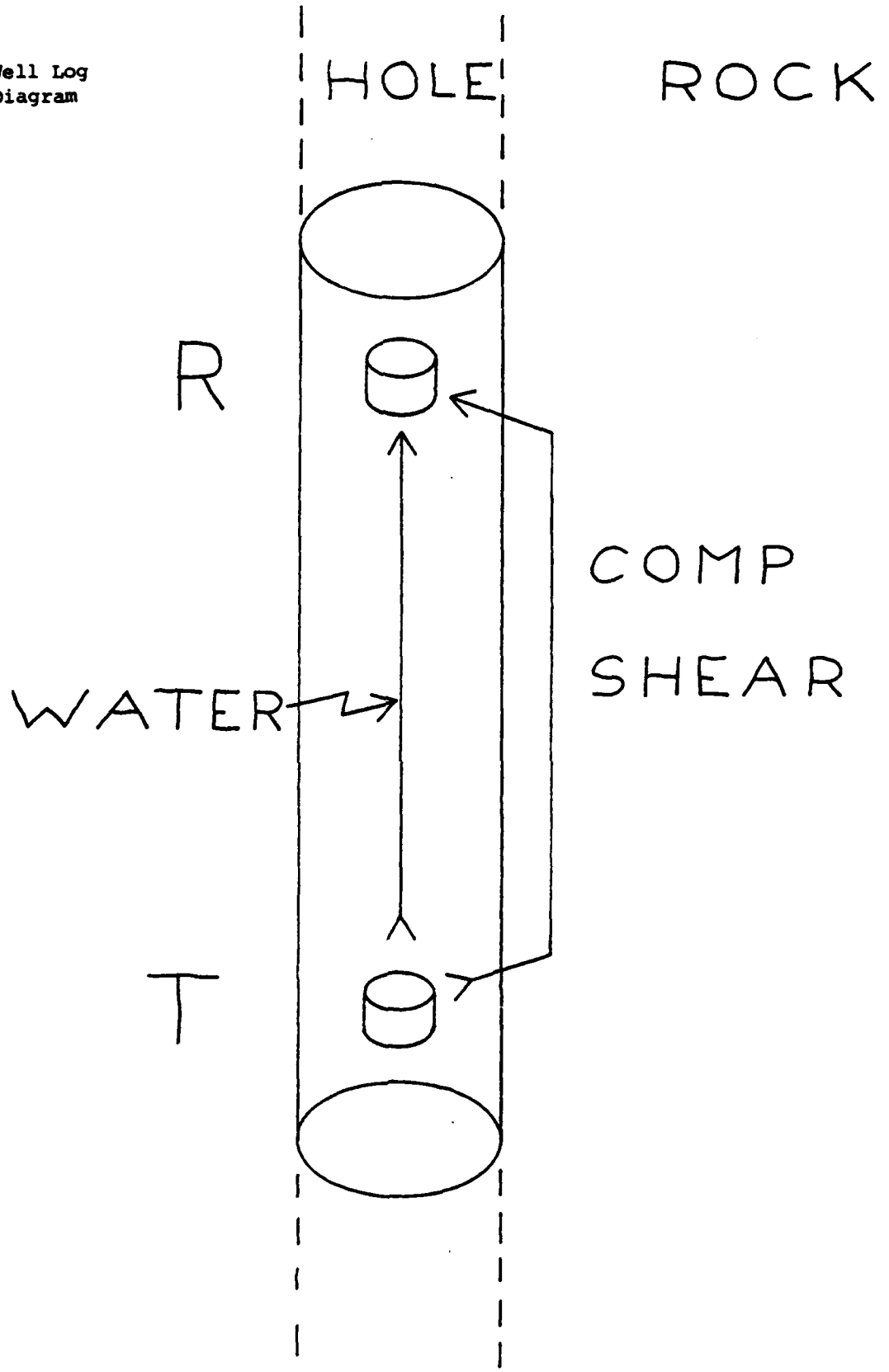


Figure 1.1

back to the top of the well. During its ascent the ultrasonic transmitter located at the bottom of the tool is pulsed (roughly once for every foot of well depth) and the pressure waveform at the receiver in the top of the tool is recorded for later analysis.

In a very simplified model of the physics of this problem, there are three paths for the ultrasonic energy to take in getting from the transmitter to the receiver, each of which has a characteristic velocity. The slowest path is that of a pressure wave traveling up the mud with which the well is packed (to prevent collapse); the medium velocity path is that of a shear wave coupled to the rock wall, and the fastest path is for the compression wave traveling up the rock wall. The quantities of interest to geologists are the velocities of the shear wave and compression wave, which could be determined by knowing their times of arrival. To ascertain the arrival times of these waves it is necessary to compress the individual events in the source data since the overlap between them is considerable.

1.1 Previous Work

Approaching the problem of event compression typically entails modeling the physical situation in an appropriate fashion and then designing an algorithm which is expected to solve the problem for data which fits that model. Subsequent

investigation of the performance of the algorithm in a realistic environment may then lead to alterations in the model, the algorithm or both.

For example, [Young,1965] analyzes the problem of detecting events in the context of RADAR. His data model is that events are separated by some minimum distance and that they are each some unknown linear combination of a set of known exponentials (the set is determined by the source waveform). In addition, he requires knowledge of the data's noise statistics. From these assumptions he derives a likelihood ratio for the beginning of an event at each point in his data (this being the event compressed signal). Unfortunately, in some situations (in particular sonic well logging) either the noise statistics are not known, detailed information about the events is not available or the number of available exponentials from which they could be composed is large making this formulation inappropriate.

A common approach to the problem of compressing seismic data is made by assuming the data was generated by convolving a fixed source wavelet with an impulsive seismic reflector series. By acquiring an accurate estimate of the source wavelet it would be possible to deconvolve and recover the reflector series (or something close to it). [Ulrych,1971] and [Tribolet,1977] have used Homomorphic

techniques to estimate the wavelet and reflector series and [Peacock,1969] used linear prediction to estimate the wavelet. Similar work has also been done on the speech pitch estimation problem [Markel,1972] by assuming that the speech signal is the convolution of a vocal tract response and a glottal excitation impulse series. The nature of these methods is that a single deconvolution operator is applied to the data in order to recover an impulsive or nearly impulsive sequence. As such these techniques will only be effective if the events in the data have similar spectra.

In this thesis we apply a recursive least square (RLS) adaptive linear prediction algorithm to event compression, using synthetic data from a data model based on an abstraction of the sonic well logging problem. Because the events in this data have independent spectra, event compression by linear time invariant filtering would not be effective (the inverse filter for each arrival would have to be different). By using an adaptive algorithm we perform time varying event compression on the input data sequence.

To implement event compression we derive two signals from the RLS algorithm, one is the post-adaption prediction error at each point and the other is a measure of the changes in the prediction coefficients as the data is processed. The two signals are compared experimentally using synthetic data

corrupted with white gaussian noise. To illustrate the impact of other processing on event compression, some experiments include prefiltering the data, decimation of the data and postfiltering the error signal.

The next chapter is an analytical presentation of the equations and issues involved in linear prediction. It starts with a discussion of linear prediction and describes the covariance method of linear prediction [Makhoul,1975]. The following section presents a derivation of the Recursive Least Squares algorithm (from the covariance method equations) and the last section of the chapter examines the problem of proper initialization of the recursion. The third chapter offers an experimental comparison of the two event compression signals mentioned above, and the thesis concludes with a discussion of the important results of these experiments and some suggestions for future work on this problem.

2. LINEAR PREDICTION

Linear prediction attempts to answer the following question: Given a sequence of data, what is the best set of p coefficients for predicting the value of each sample of the sequence with a linear combination of previous samples?[1] This formulation is presented in equation (2.1)

$$\hat{s}_k = \sum_{n=1}^{p} c_n s_{k-n} \tag{2.1}$$

where $\{s_k\}$ is the data sequence, $\{\hat{s}_k\}$ are the predictions and the c_n are the coefficients of the predictor. The criterion for determining which coefficients are best is the total squared prediction error over the chosen error region. Specifically

$$E = \sum_{k} (s_k - \hat{s}_k)^2 \qquad k \in Q \tag{2.2}$$

and the coefficients are chosen to minimize the total squared error E.

---

1. In the general case one could use an arbitrarily chosen set of previous samples, but this discussion assumes that they are the p contiguously previous samples to the one to be predicted.

The choice of the error region Q is what differentiates between the two most common methods of linear prediction. If Q extends from $-\infty$ to $+\infty$ (the data sequence is padded with zeroes or extrapolated in some other way) the resulting set of equations describe the Autocorrelation Method of linear prediction. In this case solving for the c_k involves inverting a Toeplitz matrix of autocorrelations and is usually done by some variation of Levinson's Inversion (see for example [Makhoul,1975]).

The other common method in use is the covariance method of linear prediction. As described in the next section, it requires that only the available data sequence be used to generate the predictor. Therefore the limits of Q are set by requiring that all the s_k mentioned in eqs. (2.1) and (2.2) lie in that finite set of available signal points.

2.1 The Covariance Method

The equations defining the Covariance method are:

$$\hat{s}_k = \sum_{n=1}^{p} c_n s_{k-n} \qquad (2.3)$$

$$E = \sum_{k=k_s}^{k_e} (s_k - \hat{s}_k)^2 = \text{minimum} \tag{2.4}$$

Here, p is the number of predictions coefficients and k_s and k_e are the endpoints of the prediction region Q. By defining the following structures

$$\mathbf{S} \triangleq \begin{Bmatrix} s_{k_s-1} & \cdots & s_{k_s-p} \\ \vdots & & \vdots \\ s_{k_e-1} & \cdots & s_{k_e-p} \end{Bmatrix} \tag{2.5}$$

$$\mathbf{s} \triangleq \begin{Bmatrix} s_{k_s} \\ \vdots \\ s_{k_e} \end{Bmatrix} \tag{2.6}$$

$$\mathbf{c} \triangleq \begin{Bmatrix} c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_p \end{Bmatrix} \tag{2.7}$$

the covariance method can be formulated as the following matrix projection problem.

$$\mathbf{S}\,\mathbf{c} \cong \mathbf{s} \tag{2.8}$$

Where the desired solution is that coefficient vector $\mathbf{c}$ which minimizes the distance between $\mathbf{Sc}$ and $\mathbf{s}$. The solution is any vector $\mathbf{c}$ which solves the normal equations

$$\mathbf{S}^T\mathbf{S}\,\mathbf{c} = \mathbf{S}^T\,\mathbf{s}. \tag{2.9}$$

If the matrix $\mathbf{S}^T\mathbf{S}$ is invertible then the solution is unique.

$$\mathbf{c} = (\mathbf{S}^T\mathbf{S})^{-1}\,\mathbf{S}^T\mathbf{s} \tag{2.10}$$

If there is no unique solution to eq. (2.9) ($\mathbf{S}^T\mathbf{S}$ is singular) then some restriction must be placed on $\mathbf{c}$ to make the answer unique (e.g. the order of the predictor p could be lowered

reducing the number of unknowns).

The covariance method generates a single predictor with the minimum possible total squared error over the finite prediction region Q. The RLS method is a means for calculating a sequence of covariance method predictors over successively longer prediction regions $\{Q_i\}$. It offers a computational savings compared to directly calculating the covariance method predictor for each Q_i. The derivation of this algorithm is presented in the next section.

2.2 Recursive Least Squares

The RLS algorithm finds a sequence of optimal predictors $\{\mathbf{c}[k_0]\}$ for an input signal by minimizing the total squared error from a fixed starting sample s_{k_s} up to s_{k_0}. For a signal $\{s_k\}$ and a p coefficient predictor $\mathbf{c}$, the error sequence produced by the k_0th predictor is given by

$$e_k[k_0]=s_k-\sum_{n=1}^{p} s_{k-n}c_n[k_0] \qquad k \in Q \tag{2.11}$$

and the quantity to be minimized is

$$E[k_0] = \sum_{k=k_s}^{k_0} e_k^2[k_0]. \tag{2.12}$$

This calculation could be performed by inverting a matrix for each desired predictor (see eq. 2.10). Instead, RLS does an iterative calculation which uses the previous predictor $\mathbf{c}[k_0-1]$ together with a state matrix and the new data point s_{k_0} to calculate the new predictor $\mathbf{c}[k_0]$ and the new state matrix. The advantage of this method for calculating successive predictors, over using the covariance method directly on each new prediction region, is that this calculation only requires $O(p^2)$ operations per predictor. Whereas the covariance method would need $O(p^3)$ operations per predictor for each matrix inversion.

The derivation of the iteration can be shown in matrix form as follows. Let the signal be formed into a vector and a matrix as before,

$$\mathbf{S}[k_0] \triangleq \begin{Bmatrix} s_{k_s-1} & \cdots & s_{k_s-p} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ s_{k_0-1} & \cdots & s_{k_0-p} \end{Bmatrix} \qquad \mathbf{s}[k_0] \triangleq \begin{Bmatrix} s_{k_s} \\ \vdots \\ \vdots \\ \vdots \\ s_{k_0} \end{Bmatrix}. \tag{2.13}$$

Then the error vector is

$$\mathbf{e}[k_0] \triangleq \begin{Bmatrix} e_{k_s}[k_0] \\ \cdot \\ \cdot \\ \cdot \\ e_{k_0}[k_0] \end{Bmatrix} = \mathbf{s}[k_0] - \mathbf{S}[k_0]\mathbf{c}[k_0] \tag{2.14}$$

with

$$\mathbf{c}[k_0] \triangleq \begin{Bmatrix} c_1[k_0] \\ \cdot \\ \cdot \\ \cdot \\ c_p[k_0] \end{Bmatrix} . \tag{2.15}$$

In this notation, the total error is

$$E[k_0] \triangleq \mathbf{e}^T[k_0]\mathbf{e}[k_0] \tag{2.16}$$

and the optimal choice for $\mathbf{c}[k_0]$ is

$$\mathbf{c}[k_0] = (\mathbf{S}^T[k_0]\mathbf{S}[k_0])^{-1}\mathbf{S}^T[k_0]\mathbf{s}[k_0] \tag{2.17}$$

The derivation of the RLS iteration from eq. (2.17) is performed by substituting k_0+1 for k_0 and using information available at the k_0th point to evaluate $\mathbf{c}[k_0+1]$ in an efficient manner. The new terms in the equation after the substitution are

$$\mathbf{s}[k_0+1] = \begin{Bmatrix} \mathbf{s}[k_0] \\ \text{-----} \\ s_{k_0+1} \end{Bmatrix} \tag{2.18}$$

and

$$\mathbf{S}[k_0+1] = \begin{Bmatrix} \mathbf{S}[k_0] \\ \text{----------} \\ s_{k_0} \cdot \cdot \cdot s_{k_0-p+1} \end{Bmatrix} \tag{2.19}$$

and

$$\mathbf{c}[k_0+1] = \{\mathbf{S}^T[k_0+1]\ \mathbf{S}[k_0+1]\}^{-1} \mathbf{S}^T[k_0+1]\mathbf{s}[k_0+1] \quad . \tag{2.20}$$

By introducing

$$\mathbf{r} \triangleq \begin{Bmatrix} s_{k_0} \\ \cdot \\ \cdot \\ \cdot \\ s_{k_0-p+1} \end{Bmatrix} , \tag{2.21}$$

one can substitute for terms in eqs. (2.20) using eq. (2.18) and (2.19) to get

$$\mathbf{c}[k_0+1]=\left(\mathbf{S}^T[k_0]\mathbf{S}[k_0]+\mathbf{r}\mathbf{r}^T\right)^{-1}\left(\mathbf{S}^T[k_0]\mathbf{s}[k_0] +\mathbf{r}s_{k_0+1}\right) \tag{2.22}$$

At this point introduce the following matrix identity:

> For any symmetric invertible matrix $\mathbf{V}$ and any vector $\mathbf{r}$ with the same size as a column of $\mathbf{V}$,

$$\left(\mathbf{V}^{-1} + \mathbf{r}\,\mathbf{r}^T\right)^{-1} = \mathbf{V} - \frac{\mathbf{V}\mathbf{r}\mathbf{r}^T\mathbf{V}}{1+\mathbf{r}^T\mathbf{V}\mathbf{r}} . \tag{2.23}$$

Now let

$$\mathbf{V}^{-1} \triangleq \mathbf{S}^T[k_0]\mathbf{S}[k_0] \tag{2.24}$$

and

$$\mathbf{c}' = \mathbf{c}[k_0+1] \tag{2.25a}$$

$$\mathbf{V}' = \mathbf{V}[k_0+1] = \left(\mathbf{S}^T[k_0]\mathbf{S}[k_0] + \mathbf{r}\mathbf{r}^T\right)^{-1} \tag{2.25b}$$

$$s' = s_{k_0+1} \quad . \tag{2.25c}$$

By leaving out the indices from eq. (2.22) one can write

$$\mathbf{c}' = \left(\mathbf{V} - \frac{\mathbf{V}\mathbf{r}\mathbf{r}^T\mathbf{V}}{1+\mathbf{r}^T\mathbf{V}\mathbf{r}}\right)(\mathbf{S}^T\mathbf{s}+\mathbf{r}s') \quad . \tag{2.26}$$

Rearranging terms and combining with eq. (2.17) leads to an expression which describes how to update the predictor.

$$\mathbf{c}' = \mathbf{c} + \frac{\mathbf{V}\mathbf{r}}{1+\mathbf{r}^T\mathbf{V}\mathbf{r}}(s'-\mathbf{r}^T\mathbf{c}) \tag{2.27}$$

The other RLS equation is derived by combining eqs. (2.23), (2.24) and (2.25b) and it shows how to update the state matrix **V**.

$$\mathbf{V}' = \mathbf{V} - \frac{\mathbf{V}\mathbf{r}\mathbf{r}^T\mathbf{V}}{1+\mathbf{r}^T\mathbf{V}\mathbf{r}} \tag{2.28}$$

The full RLS iteration procedes as follows:

Starting with $\mathbf{V}$ and $\mathbf{c}$ from the previous iteration and the vector of previous signal points

$$\mathbf{r} \triangleq \begin{Bmatrix} s_{k_0} \\ \cdot \\ \cdot \\ \cdot \\ s_{k_0-p+1} \end{Bmatrix} , \tag{2.29}$$

use the new signal point to calculate the new predictor

$$\mathbf{c}' = \mathbf{c} + \frac{\mathbf{Vr}}{1+\mathbf{r}^T\mathbf{Vr}}(s'-\mathbf{r}^T\mathbf{c}) \quad . \tag{2.30}$$

Finally, calculate the new state matrix

$$\mathbf{V}' = \mathbf{V} - \frac{\mathbf{Vrr}^T\mathbf{V}}{1+\mathbf{r}^T\mathbf{Vr}} \quad . \tag{2.31}$$

2.2.1 Initialization of RLS

The RLS algorithm offers a means to efficiently extend the error region of a covariance method predictor by one point. Thus, given an initial set of prediction coefficients $\mathbf{c}_0$ and an initial state matrix $\mathbf{V}_0$, one can recursively calculate the coefficients of the predictors of all possible error region extents up to the end of the data. However, calculating the initial covariance predictor normally involves a matrix inversion which can be computationally costly. In addition, to perform both an initial matrix inversion for a small data interval and then execute the RLS iteration for the remaining data would involve a large amount of program code, since those two tasks involve fundamentally different calculations. It was desirable to find an alternative means for initializing the RLS algorithm which did not involve an initial covariance predictor calculation. Examining a first order case suggests a solution.

For a first order (single coefficient) predictor, the main variables have the following form:

$$\mathbf{c}[k_0] = c_1[k_0] \tag{2.32a}$$

$$\mathbf{S}[k_0] = \begin{Bmatrix} s_{k_s-1} \\ \cdot \\ \cdot \\ \cdot \\ s_{k_0-1} \end{Bmatrix} \tag{2.32b}$$

$$\mathbf{s}[k_0] = \begin{Bmatrix} s_{k_s} \\ \cdot \\ \cdot \\ \cdot \\ s_{k_0} \end{Bmatrix} \tag{2.32c}$$

and the predictor which solves the covariance method equation (eq. 2.22) is given by

$$c_1[k_0] = \frac{\sum_{k=k_s-1}^{k_0-1} s_k s_{k+1}}{\sum_{k=k_s-1}^{k_0-1} s_k^2} \tag{2.33}$$

Because the variables c, v and r are now one dimensional, the equations for the RLS iteration from (2.30) and (2.31) are

$$c' = c + \frac{vr}{1+r^2v}\{s'-rc\} \tag{2.34a}$$

or

$$c' = \frac{\frac{c}{v} + rs'}{\frac{1}{v} + r^2} \tag{2.34b}$$

and

$$v' = v - \frac{v^2r^2}{1+r^2v} \tag{2.35a}$$

or

$$v' = \frac{1}{\frac{1}{v} + r^2} \tag{2.35b}$$

where

$$r \triangleq s_{k_0} \tag{2.36}$$

and

$$v^{-1} \triangleq \mathbf{S}^T[k_0]\,\mathbf{S}[k_0] = \sum_{k=k_s-1}^{k_0-1} s_k^2 \tag{2.37}$$

Suppose one started the iteration of equations (2.34) and (2.35) at the first point in the error region (i.e. $k_0+1=k_s$). By assigning the initial values $v=v_{init}$ $c=c_{init}$ we have

$$v = \frac{1}{\frac{1}{v_{init}} + \sum_{k=k_s-1}^{k_0-1} s_k^2} \tag{2.38}$$

Clearly, for this iteration to conform to the requirement of eq. (2.37), $v_{init}=\infty$. Given that result, the successive values of c are

$$c[k_0] = \frac{\frac{c_{init}}{v_{init}} + \sum_{k=k_s-1}^{k_0-1} s_k s_{k+1}}{\frac{1}{v_{init}} + \sum_{k=k_s-1}^{k_0-1} s_k^2} \tag{2.39}$$

Therefore, given $v_{init}=\infty$, the subsequent values of c solve eq. (2.33) making them identical to the equivalent covariance method predictors (irrespective of the initial value of c). This behavior is shown experimentally in figure (2.1). The data comes from exciting a 1-pole linear system with an impulse at n=10. The system function and time response were

$$H(z) = \frac{1}{1-\frac{.98}{z}}$$

$$s[n] = u_{-1}[n-10]\ (.98)^{n-10}$$

The covariance method in the absence of noise should predict this signal exactly when given more than one point of its impulse response in the chosen error region. As can be seen from the figure, RLS also predicts the response for every point after the first. This method of initialization is an effective means for matching the RLS predictors to tnose generated by the covariance method.

2.2.2 Higher Order Initialization

When a multicoefficient predictor is used, the mathematical approach used in the previous section to find $\mathbf{V}_{init}$ and $\mathbf{c}_{init}$ is not fruitful since the vector matrix products do not commute and cancel in the fashion of eqs. (2.34) and (2.35). One reason for the difficulty of choosing the initial values of **c** and **V** in this circumstance is that the covariance predictor for which we aim is not uniquely specified until the error region is at least p points in length; and even then only if the data sequence requires at

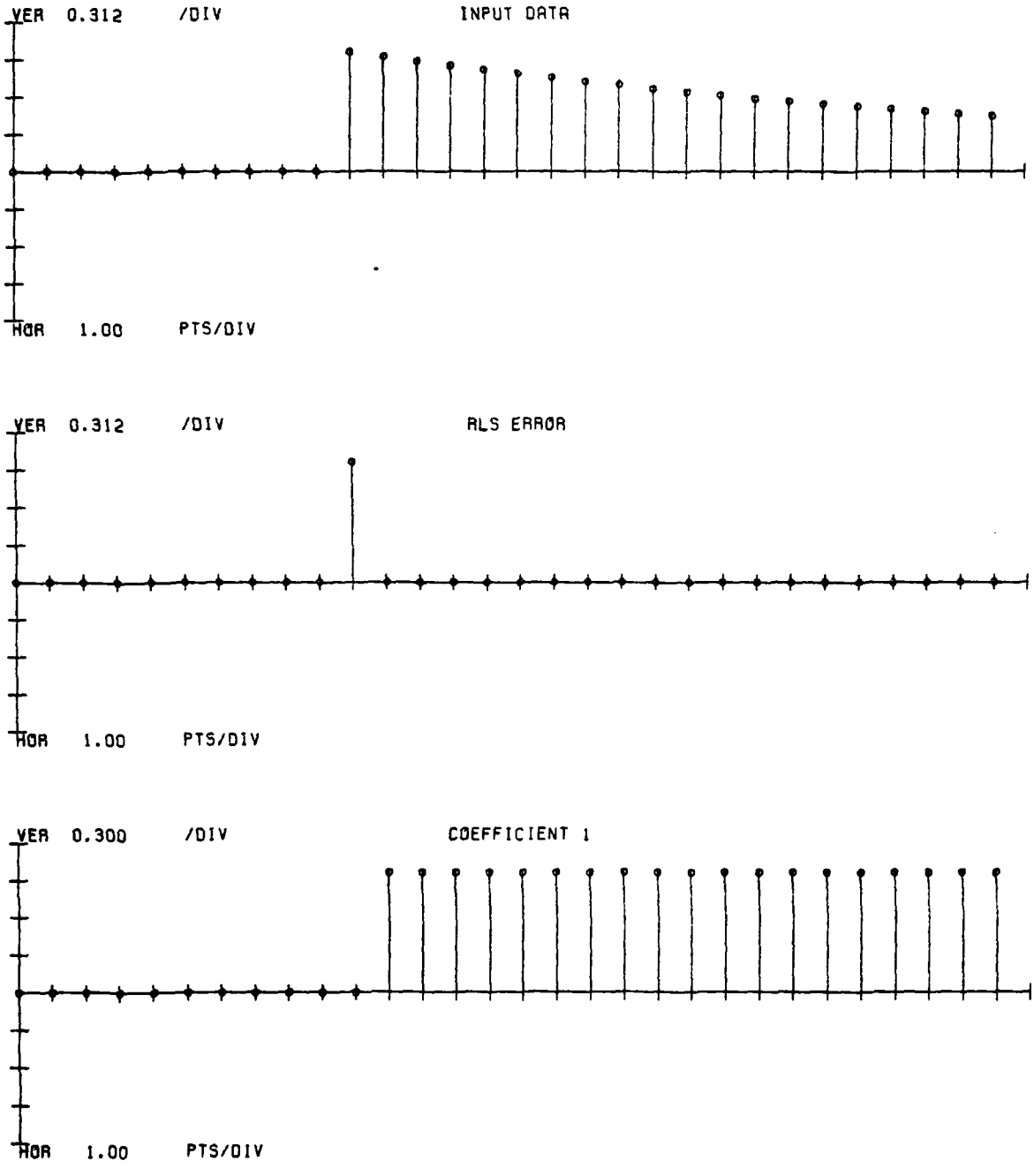

VER 0.312 /DIV
INPUT DATA
HOR 1.00 PTS/DIV
VER 0.312 /DIV
RLS ERROR
HOR 1.00 PTS/DIV
VER 0.300 /DIV
COEFFICIENT 1
HOR 1.00 PTS/DIV

Figure 2.1

least a p pole model to describe it. Consider the covariance matrix definition

$$\mathbf{R} \triangleq \mathbf{S}^T \mathbf{S} \tag{2.40}$$

The iteration for $\mathbf{R}$ is

$$\mathbf{R}[k+1] = \mathbf{R}[k] + \mathbf{r}[k]\ \mathbf{r}^T[k] \qquad k \geq k_s - 1 \tag{2.41}$$

$$\mathbf{R}[k_s - 1] = \mathbf{0} \tag{2.42}$$

therefore,

$$\mathbf{R}[k_0] = \sum_{k=k_s-1}^{k_0-1} \mathbf{r}\mathbf{r}^T \qquad k_0 > k_s \tag{2.43}$$

Now consider the iteration for $\mathbf{V}$ along with $\mathbf{V}_{init} = \mathbf{I}\alpha$ as an approximation for $\mathbf{R}^{-1}$. If $\hat{\mathbf{R}} = \mathbf{V}^{-1}$ then given equation (2.25b)

$$\hat{\mathbf{R}}[k_s] = \frac{1}{\alpha}\mathbf{I} + \mathbf{r}[k_s - 1]\ \mathbf{r}^T[k_s - 1] \tag{2.44}$$

and in general

$$\hat{\mathbf{R}}[k_0] = \frac{1}{\alpha}\mathbf{I} + \sum_{k=k_s-1}^{k_0-1} \mathbf{r}[k]\mathbf{r}^T[k] \qquad (2.45)$$

Therefore setting $\alpha=\infty$ will make $\hat{\mathbf{R}}=\mathbf{R}$ and as in the first order case the RLS predictor will correspond to the covariance method predictor.

In the first order case the initial $\mathbf{c}$ was irrelevent. Ultimately, the same is true in the multidimensional case as well since $\mathbf{c}$ solves

$$\mathbf{R}\,\mathbf{c} = \mathbf{S}^T\,\mathbf{s} \qquad (2.46)$$

and once $\mathbf{R}$ is invertible, the value for $\mathbf{c}$ is fully determined. However, for the first few points of the iteration, $\mathbf{R}$ is singular and the trajectory followed by $\mathbf{c}$ depends on the data values and the value of$\mathbf{c}_{init}$. As an example, figures 2.2 and 2.3 show a 4-pole impulse response beginning at sample 20 which was processed by a 4-coefficient predictor with the initialization

$$V_{init} = 10^{10} I$$

$$c_{init} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

The predictor used in figure 2.2 is started at k_s=10; the one in figure 2.3 is started at k_s=30 (thus these examples use different initial data values). As can be seen, both predictors reach the same value after 4 steps into the non-zero data, but their trajectories differ. To illustrate the dependence of trajectory on c_{init}, figure 2.4 presents a 4-coefficient predictor started with

$$V_{init} = 10^{10} I$$

$$c_{init} = \begin{Bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{Bmatrix}$$

$$k_s = 10$$

Again the same predictor value is reached after 4 steps into the data, but the trajectory differs from the previous cases.

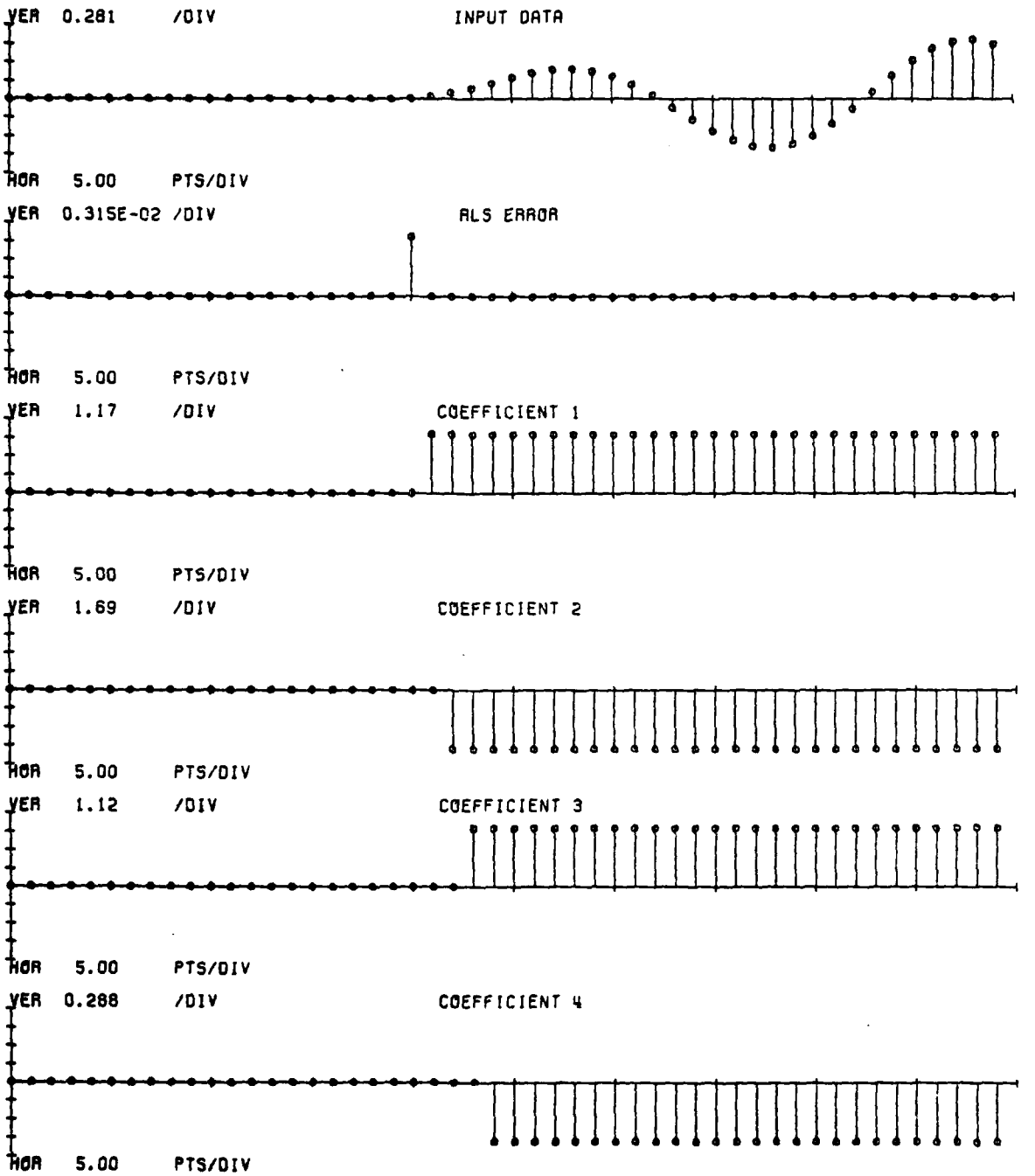

VER 0.281 /DIV
INPUT DATA
HOR 5.00 PTS/DIV
VER 0.315E-02 /DIV
RLS ERROR
HOR 5.00 PTS/DIV
VER 1.17 /DIV
COEFFICIENT 1
HOR 5.00 PTS/DIV
VER 1.69 /DIV
COEFFICIENT 2
HOR 5.00 PTS/DIV
VER 1.12 /DIV
COEFFICIENT 3
HOR 5.00 PTS/DIV
VER 0.288 /DIV
COEFFICIENT 4
HOR 5.00 PTS/DIV

Figure 2.2

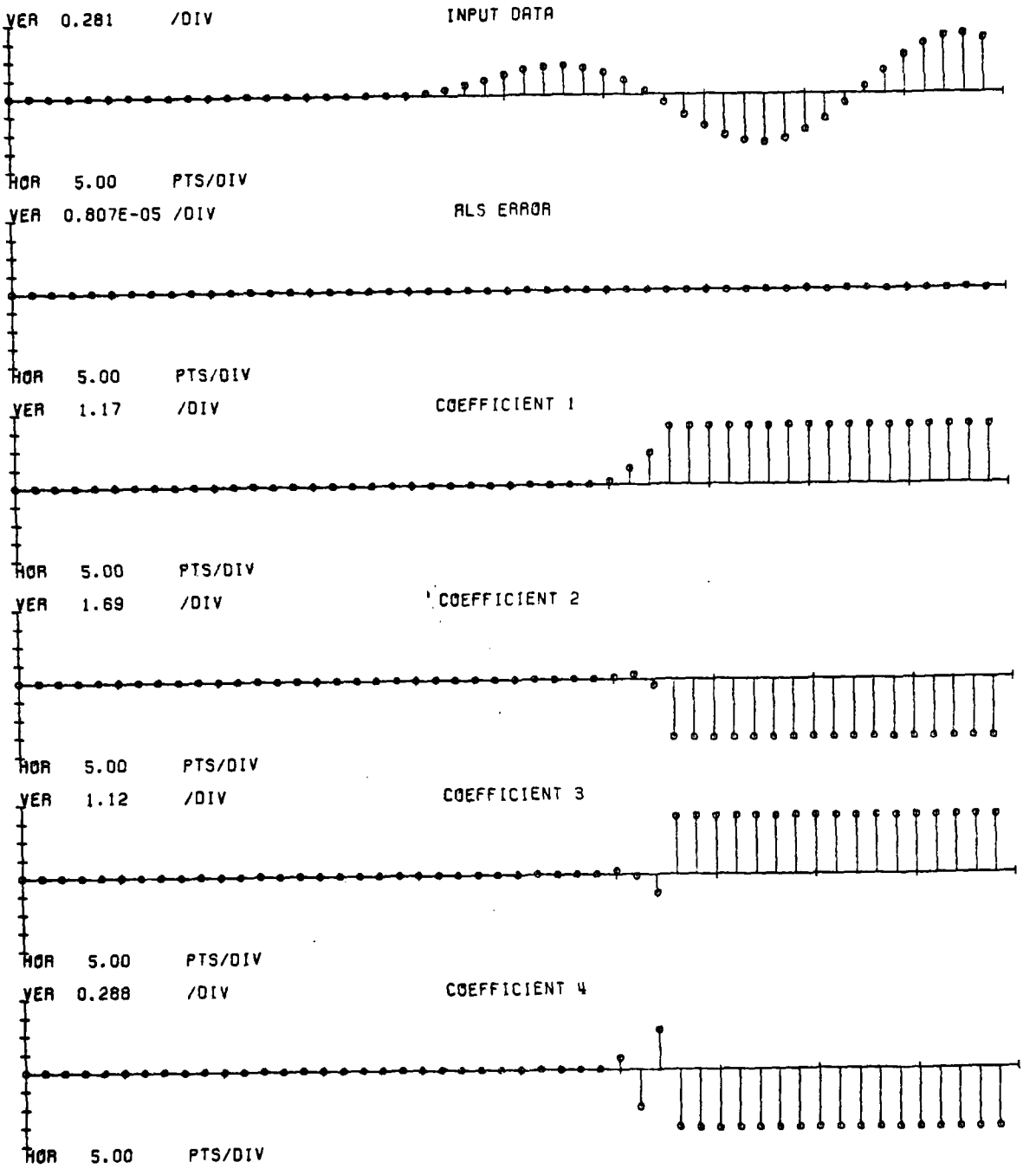

VER 0.281 /DIV
INPUT DATA
HOR 5.00 PTS/DIV
VER 0.807E-05 /DIV
RLS ERROR
HOR 5.00 PTS/DIV
VER 1.17 /DIV
COEFFICIENT 1
HOR 5.00 PTS/DIV
VER 1.69 /DIV
COEFFICIENT 2
HOR 5.00 PTS/DIV
VER 1.12 /DIV
COEFFICIENT 3
HOR 5.00 PTS/DIV
VER 0.288 /DIV
COEFFICIENT 4
HOR 5.00 PTS/DIV

Figure 2.3

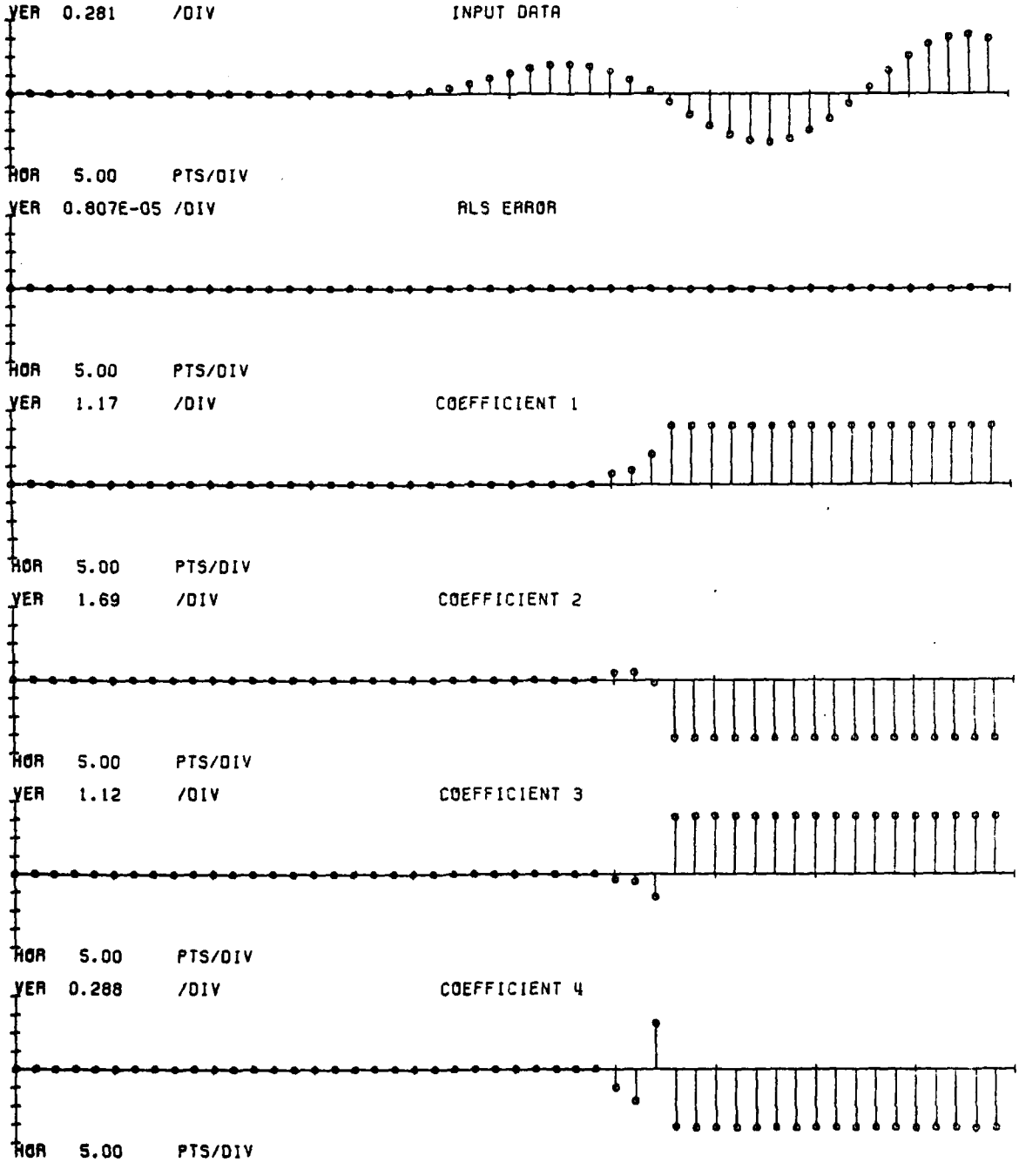

VER 0.281 /DIV
INPUT DATA
HOR 5.00 PTS/DIV
VER 0.807E-05 /DIV
RLS ERROR
HOR 5.00 PTS/DIV
VER 1.17 /DIV
COEFFICIENT 1
HOR 5.00 PTS/DIV
VER 1.69 /DIV
COEFFICIENT 2
HOR 5.00 PTS/DIV
VER 1.12 /DIV
COEFFICIENT 3
HOR 5.00 PTS/DIV
VER 0.288 /DIV
COEFFICIENT 4
HOR 5.00 PTS/DIV

Figure 2.4

2.2.3 Numerical Considerations

The initialization derived in the last section offers potential numerical problems. The first few iterations of these equations have great potential for numerical error since the associated **R** matrix is clearly singular. In our initial work we discovered a threshold of about 10^{10} for the value of α in eq. (2.44) above which the iteration would not cause any change in the coefficients as the data was processed. Since it is necessary to choose $1/\alpha$ so that the smallest non-zero eigenvalue of **R** is large in comparison, it is undesirable to require that the value for α be much smaller than 10^{10}. Fortunately, by using double precision arithmetic we were able to succesfully use values exceeding 10^{20} for initialization. Subsequent comparisons of the direct covariance method calculation with the RLS algorithm initialized at

$$\mathbf{V}_{init} = 10^{10}\mathbf{I}$$

$$\mathbf{c}_{init} = \mathbf{0}$$

showed coefficient differences well under one part in 10^{-5} for data lengths up to 1024. That accuracy was deemed sufficient for this work, but for a more critical application it would be

wise to determine specifically how the roundoff error of the computer affects the initialization of the algorithm.

2.3 Discussion

This chapter has presented the basic mathematics behind the RLS algorithm including some simple examples of its behavior. For more information on the topic of linear prediction the reader is invited to examine [Makhoul,1975] and the references he cites. More information about RLS in particular can be found in [Eykoff,1974] and more recent ladder forms of recursive linear prediction are illustrated in [Satorious,1979].

3. EVENT COMPRESSION WITH RLS

The previous chapter described the RLS method of linear prediction and how it can be initialized and used to generate a series of predictors from an input data sequence. This chapter examines how RLS can be used to compress events. We begin by proposing a model for multiple event signals which is a very simple abstraction of the sonic well log situation described in the introduction. We then describe two event compressed signals which can be extracted from the RLS iteration and finally experiments are performed to illustrate the performance of these signals.

The concept behind using an adaptive prediction algorithm like RLS for event compression is the following. Consider the series of predictors created by the RLS algorithm as a single time-varying predictor which minimizes the total prediction error energy over an expanding region. When processing an input sequence containing distinct events, one expects the predictor to make errors at the beginning of each event since the beginning of the event will not be predictable by the algorithm. This error burst will be accompanied by a change in the predictor coefficients as the algorithm adapts to predict the event. Hopefully, after the first few points of the event have passed, the error pulse will die away and the predictor will stop changing. If that is the case, then

the prediction error and the coefficient changes would both respond to the events in a way that could be used to perform event compression of the input data.

3.1 The Data Model

This thesis was motivated by the problem of sonic well logging which was presented in the introduction. The signals present in a real well log are very complex do to the geometry of the well. Rather than attempting to accurately model the well log data (a complicated task in itself) we chose to fabricate synthetic data which had the appearence of a well log and exhibited what we felt was an important feature of well logs; namely that the signal contain multiple bursts with independent spectra. The data model for the signals used in this study is shown in figure 3.1. A single impulse is fed to each of three delays D_1-D_3. Their outputs drive the discrete-time all-pole systems H_1-H_3 generating three delayed pulses which are added to produce the signal s[k]. We chose a three pulse model because the problem of seismic well logging can be considered a three pulse problem.

To make it possible to compare the various figures presented later, a small set of representative signals was chosen from the data model given above to perform the experiments. Most use either a signal labeled Burst1 or some

THE DATA MODEL

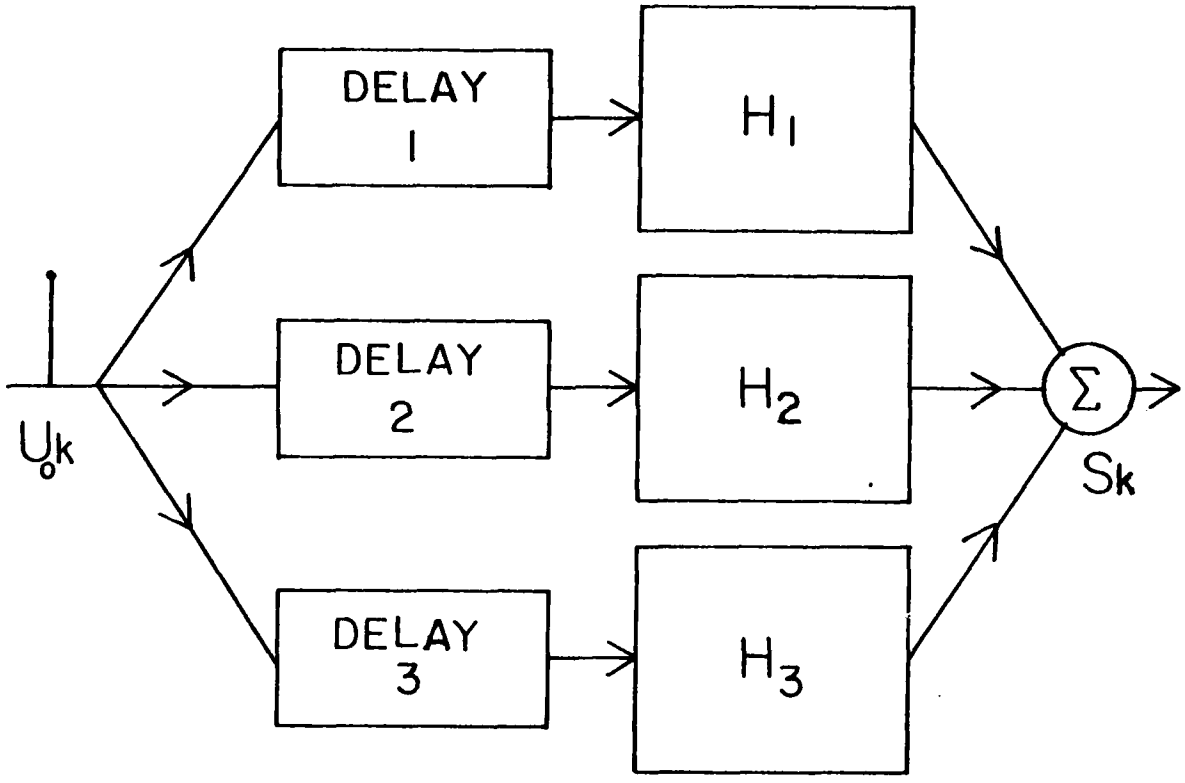


Figure 3.1

combination of its component pulses Burstla, Burstlb or Burstlc. The parameters used to generate Burstl were:

Burstla:

$$D_1 = 50 \text{ points} \qquad H_1 = \frac{G_1}{1-3.73z^{-1}+5.4z^{-2}-3.58z^{-3}+.92z^{-4}}$$

Burstlb:

$$D_2 = 150 \text{ points} \qquad H_2 = \frac{G_2}{1-3.89z^{-1}+5.74z^{-2}-3.81z^{-3}+.96z^{-4}}$$

Burstlc:

$$D_3 = 250 \text{ points} \qquad H_3 = \frac{G_3}{1-1.92z^{-1}+.98z^{-2}}$$

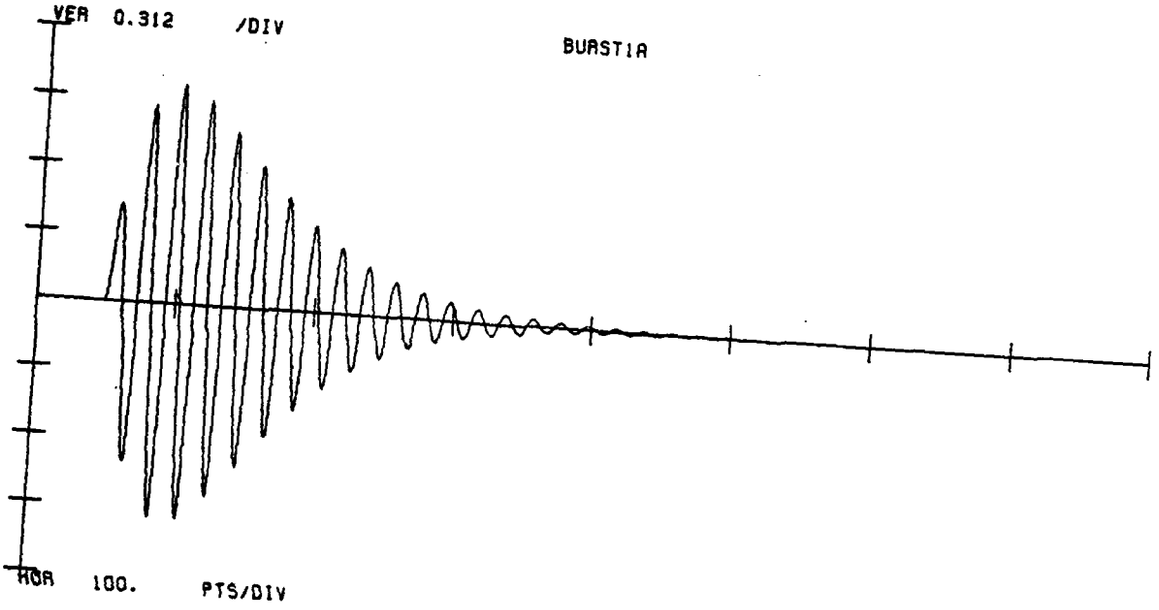
where the gains were chosen to give the component arrivals peak powers of 1, .5, and .2 respectively. Figures 3.2 through 3.5 show the time response and log magnitude spectra for these signals.[1]

The first two bursts have 2 superimposed pairs of complex poles as indicated by their gradual build up in amplitude. Whereas the third burst is due to a single pair of

---

1. The graphs in these figures and in most of those following have been linearly interpolated.

complex poles giving it a much sharper onset. These choices were made to give the data the appearance of a sonic well log. In particular, the slow growth of the second burst is a common feature of well logs and makes finding it either by event compression or by eye a difficult task.


VER 0.312 /DIV
BURSTIA
HOR 100. PTS/DIV

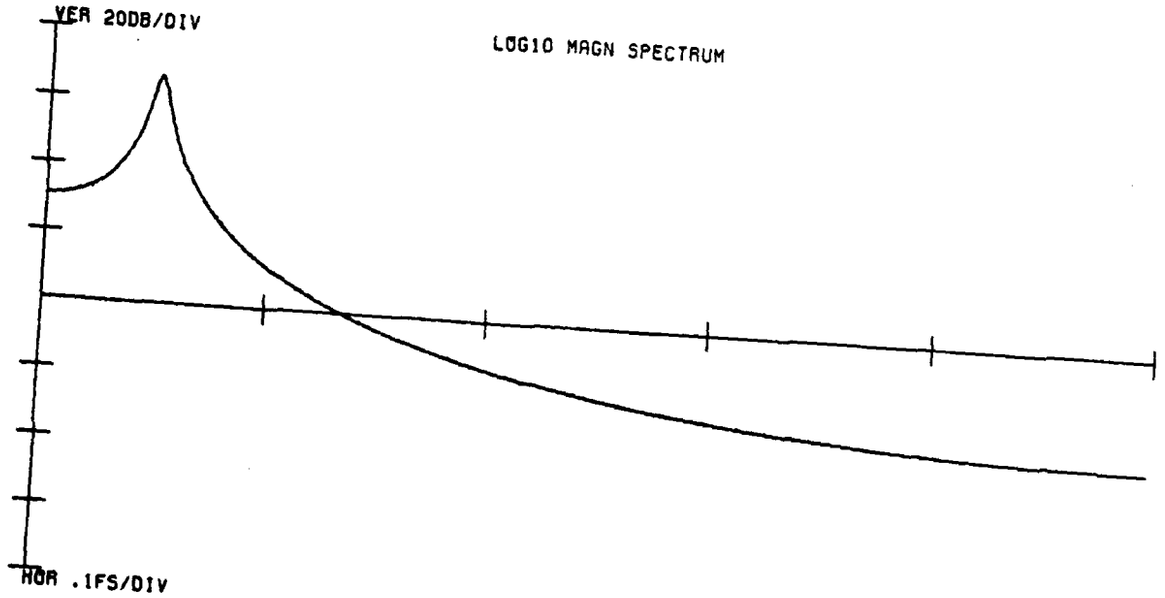
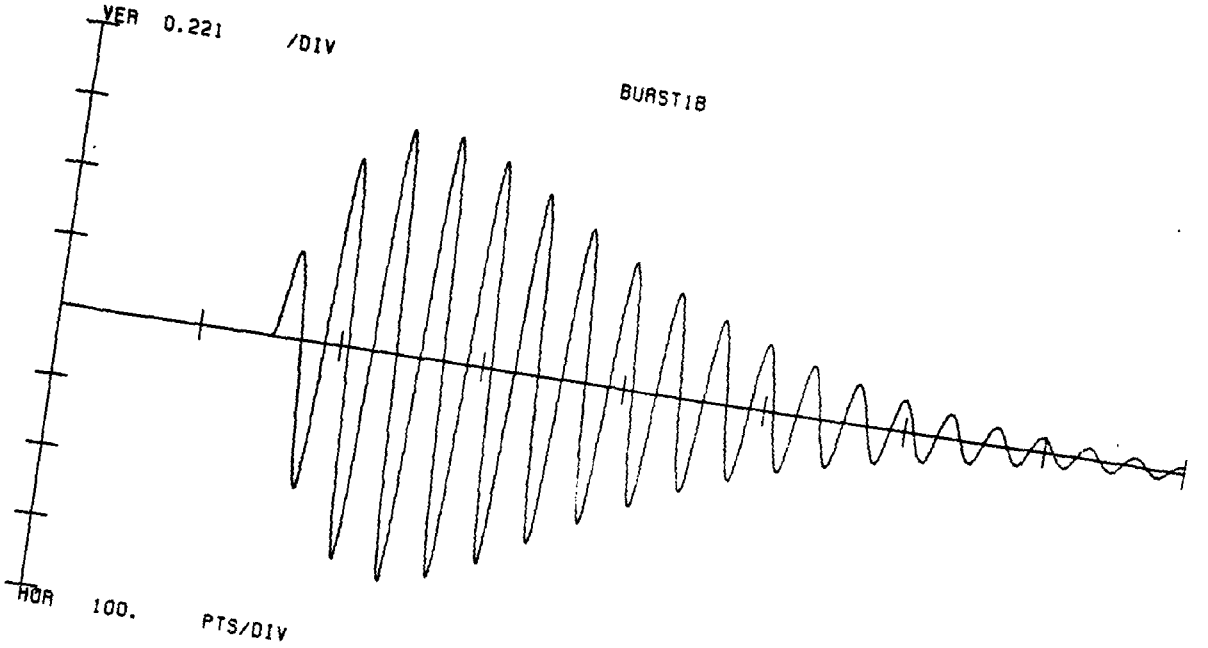

VER 20DB/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.2


VER 0.221 /DIV
BURST1B
HOR 100. PTS/DIV

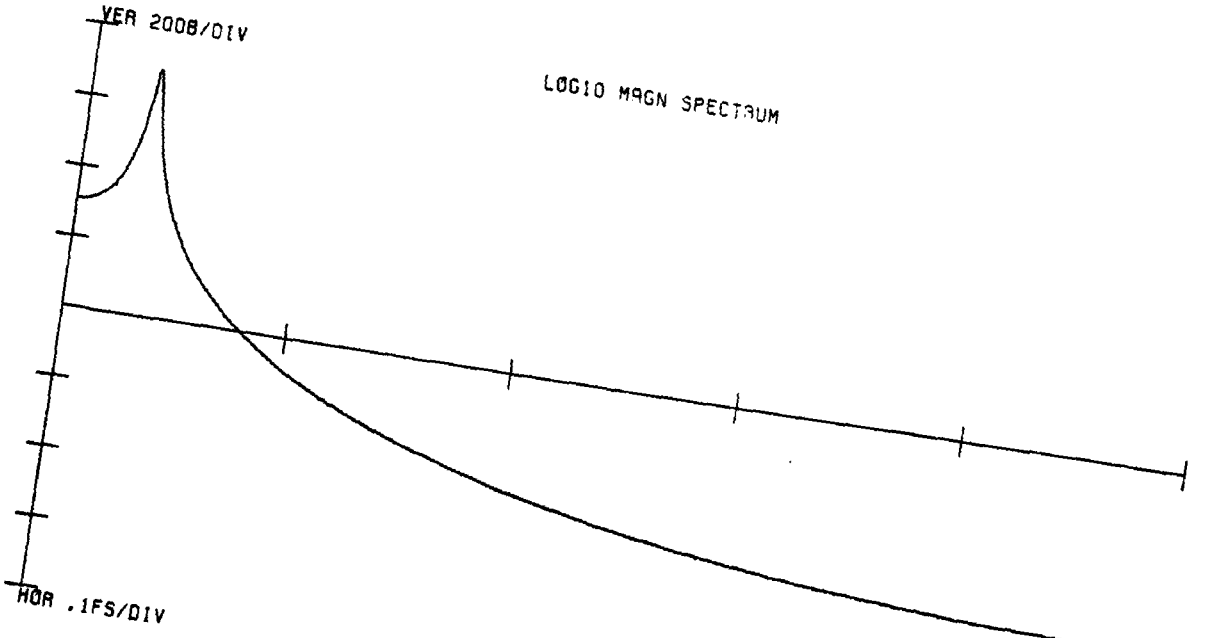

VER 2008/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.3

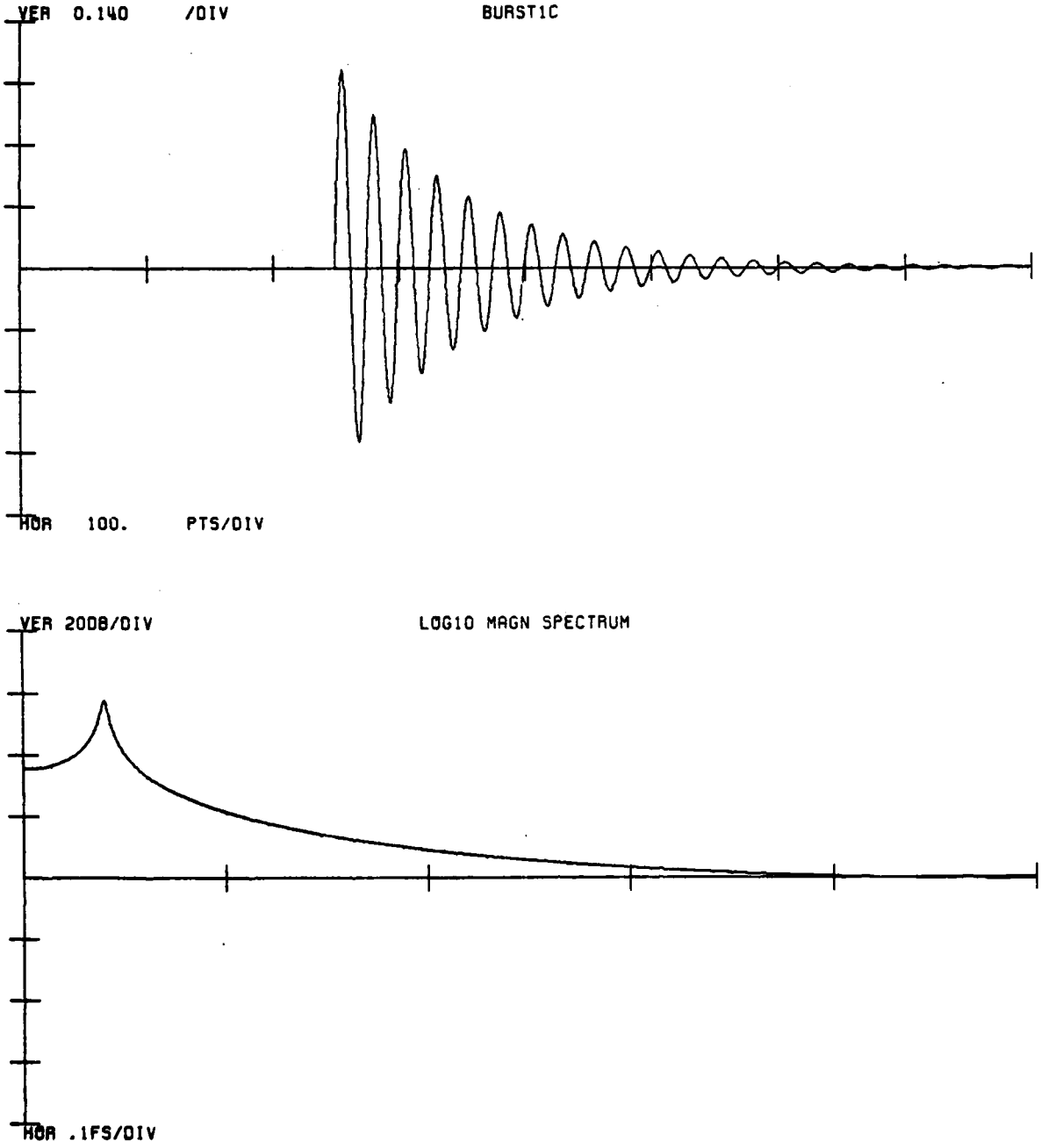
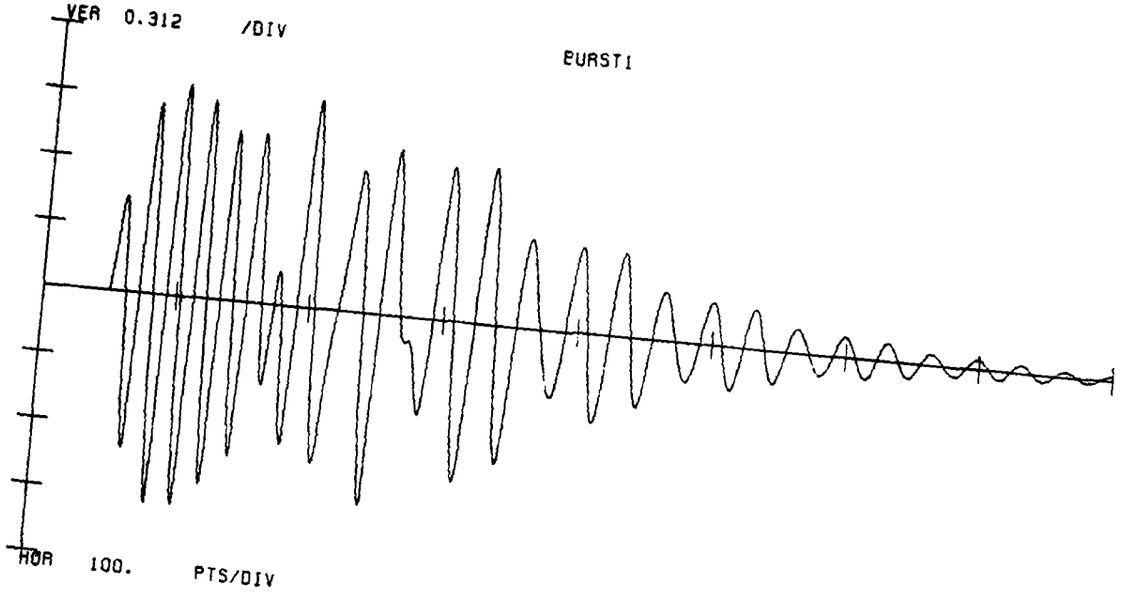

VER 0.140 /DIV
BURST1C
HOR 100. PTS/DIV
VER 20DB/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.4


VER 0.312 /DIV
BURST1
HOR 100. PTS/DIV

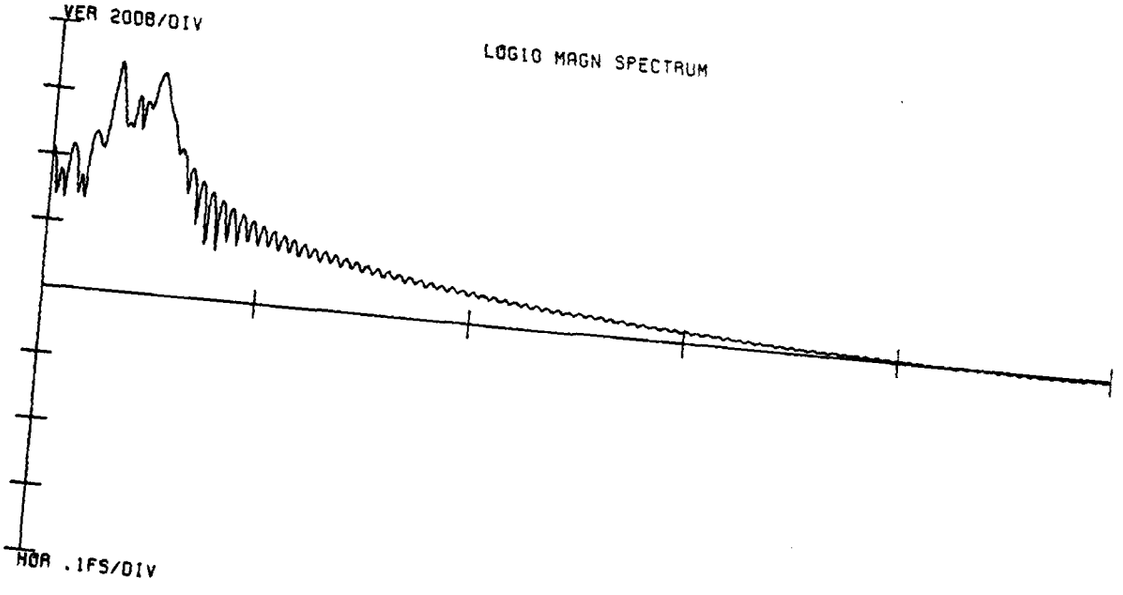

VER 2008/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.5

3.2 Event Compression Signals

There are three signals available from the RLS iteration which we examined for potential use in event compression. These are the predictor coefficient vector **c**, the prediction error at the new point before the update e_b, and the prediction error at the new point after the update e_a. In terms of the equations for the update given in the last chapter (eqs. 2.30 and 2.31) these error sequences are defined as

$$e_b[k_0+1] = s[k_0+1] - \mathbf{r}^T\mathbf{c} \tag{3.1}$$

and

$$e_a[k_0+1] = s[k_0+1] - \mathbf{r}^T\mathbf{c}' \tag{3.2}$$

Of the two error signals we chose only to work with the post update prediction error e_a because it contained more compressed events as illustrated in figures 3.6 and 3.7. The data sequence for these figures was a single pole burst and as expected both error signals have pulses at the first point of the burst (the first point of the burst can not be predicted with RLS since the previous points are all zero). However, though the second point of the burst is predictable from the first, only e_a is zero for that point since e_b is calculated

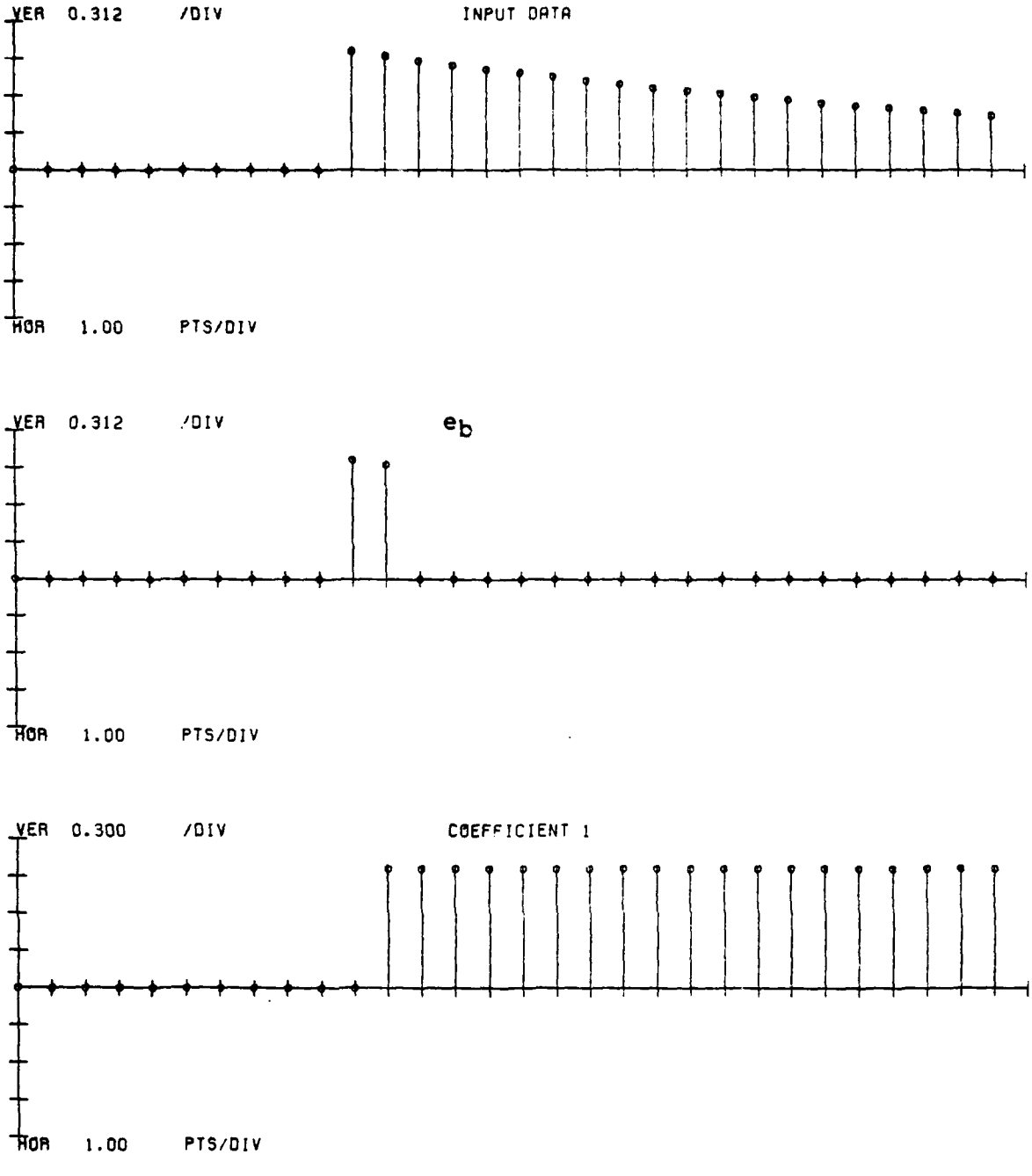

VER 0.312 /DIV
INPUT DATA
HOR 1.00 PTS/DIV
VER 0.312 /DIV
e_b
HOR 1.00 PTS/DIV
VER 0.300 /DIV
COEFFICIENT 1
HOR 1.00 PTS/DIV

Figure 3.6

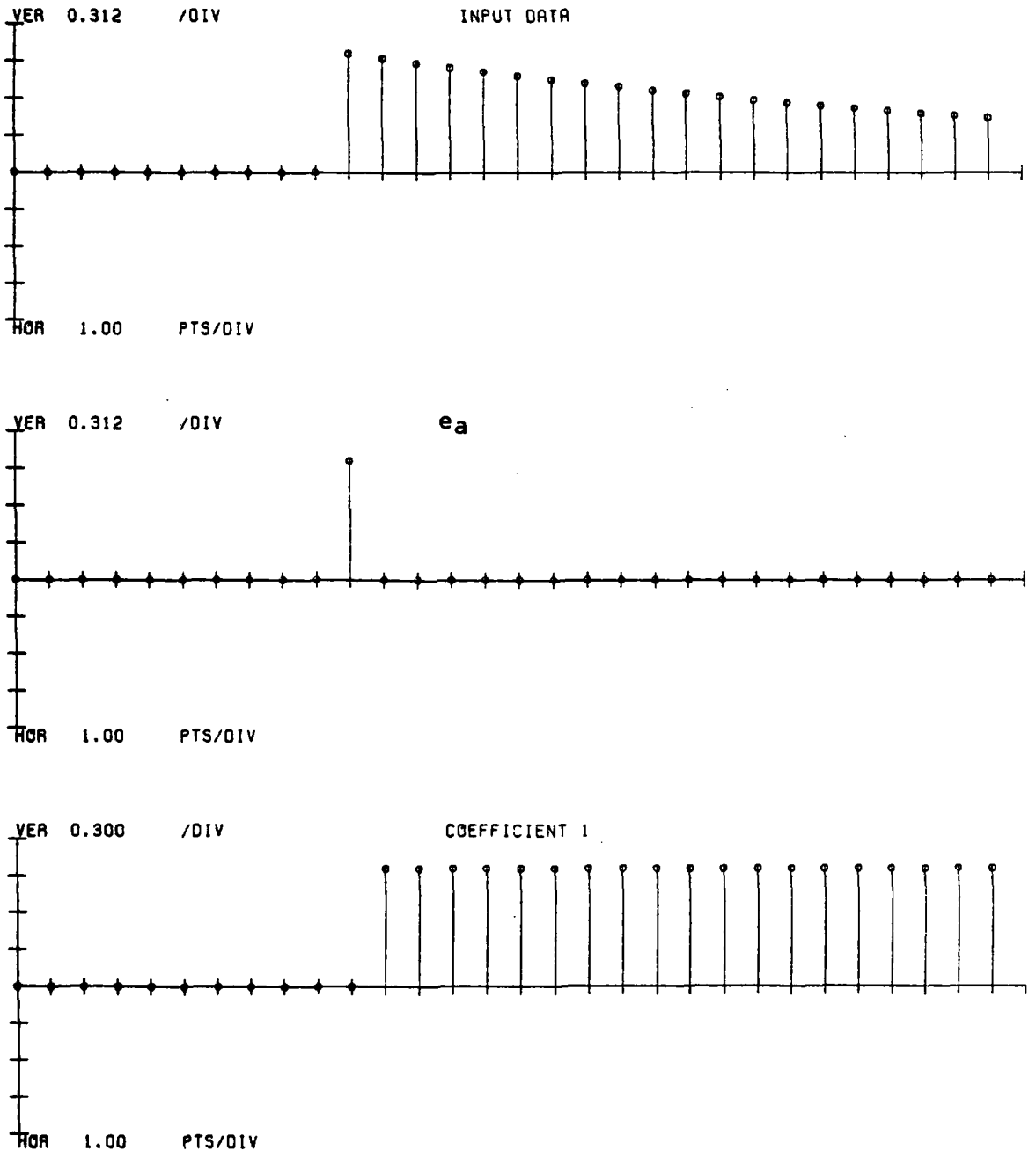

VER 0.312 /DIV
INPUT DATA
HOR 1.00 PTS/DIV
VER 0.312 /DIV
e_a
HOR 1.00 PTS/DIV
VER 0.300 /DIV
COEFFICIENT 1
HOR 1.00 PTS/DIV

Figure 3.7

before updating the predictor. This effect leads to longer error bursts in e_b at events in the data and led to our choice to use e_a for our work. All further references to the RLS error or the prediction error in this thesis are to e_a unless otherwise noted.

One final point should be noted about this error signal, particularly when compared to the error sequence generated by the covariance method on data of this type. The covariance method error sequence comes from applying a single predictor to the entire signal, whereas the RLS error comes from applying a different predictor to each point in the signal. Because the RLS predictor need only minimize the error energy to the left of the predicted point and not over the entire sequence (as is the case with the covariance method), the RLS error sequence will almost always have lower total energy than the error sequence of the covariance method for the same predictor order. As a matter of observation, we have found that this reduction in energy takes the form of shorter error bursts at the events in the data; though as yet we have not proven that this must be the case.

In addition to the prediction error signal, we examined the changes in the predictor coefficients as a means of generating an event compressed signal. This idea stemmed from observing how rapidly the prediction coefficients settled

after a new event occurred.

Figure 3.8 demonstrates the behavior of a 12 coefficient RLS predictor on the signal Burst1 defined in the last section (some of the coefficients are omitted due to the lack of space). The first event (burst) at point 50 causes a single non-zero error point and a rapid change in the predictor. This behavior is expected since the signal as of the first event is all-pole. The second event (point 150) causes very little error or coefficient change (presumably because of the similarity between the first and second bursts). But note the activity at the third event (point 250). Both the error and the coefficients settle in a short time compared to the burst, despite the fact that the signal is no longer all-pole. The rapid settling time of the coefficients led us to formulate a signal based on them which could be used for event compression.

A coefficient change signal was generated by low pass filtering each $c_i[k]$ with a single pole low pass filter to produce the vector $\bar{\mathbf{c}}[k]$ and measuring the distance $||\mathbf{c}[k]-\bar{\mathbf{c}}[k]||$. Small random variations in $\mathbf{c}[k]$ about a fixed value are reflected in the coefficient change signal as noise, with each sample having a height equal to the radial distance from the average predictor $\bar{\mathbf{c}}[k]$ to the instantaneous predictor $\mathbf{c}[k]$. However, when $\mathbf{c}[k]$ jumps to a new value due to an event

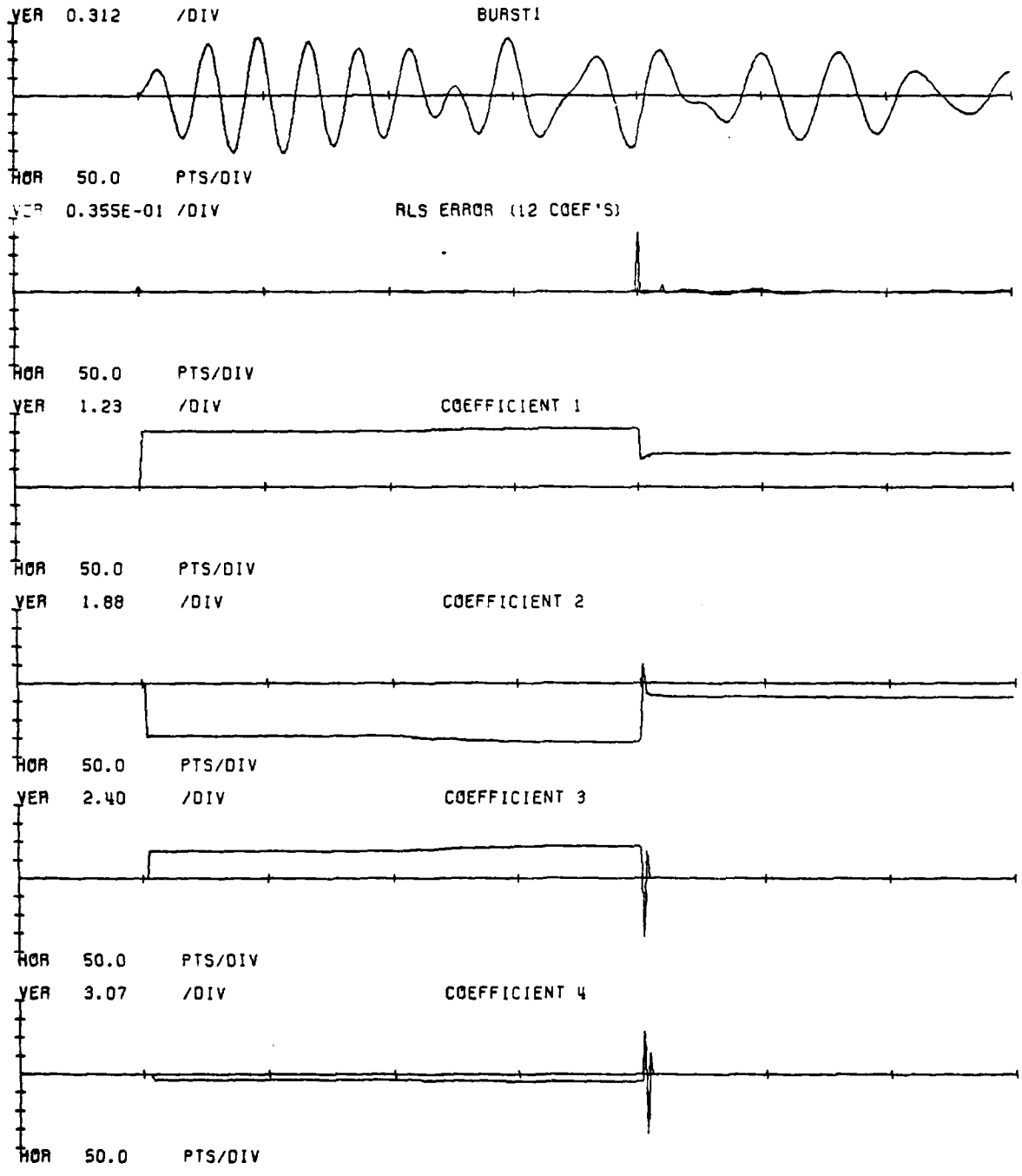

VER 0.312 /DIV
BURST1
HOR 50.0 PTS/DIV
VER 0.355E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 1.23 /DIV
COEFFICIENT 1
HOR 50.0 PTS/DIV
VER 1.88 /DIV
COEFFICIENT 2
HOR 50.0 PTS/DIV
VER 2.40 /DIV
COEFFICIENT 3
HOR 50.0 PTS/DIV
VER 3.07 /DIV
COEFFICIENT 4
HOR 50.0 PTS/DIV

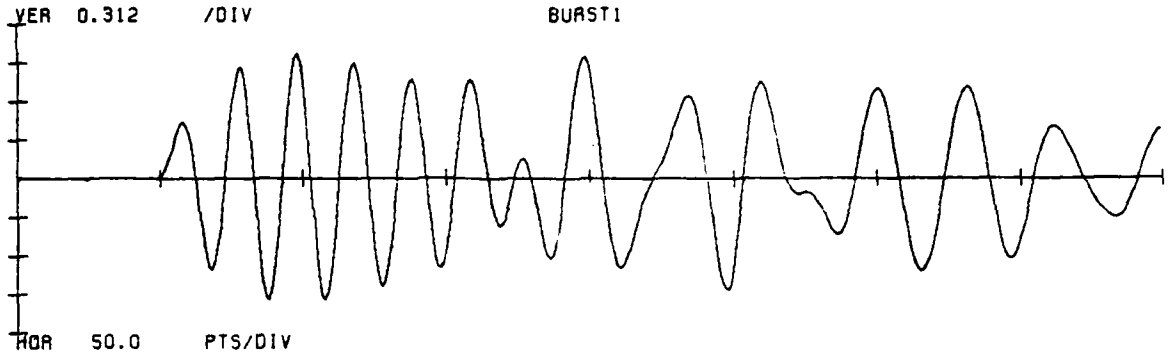
Figure 3.8

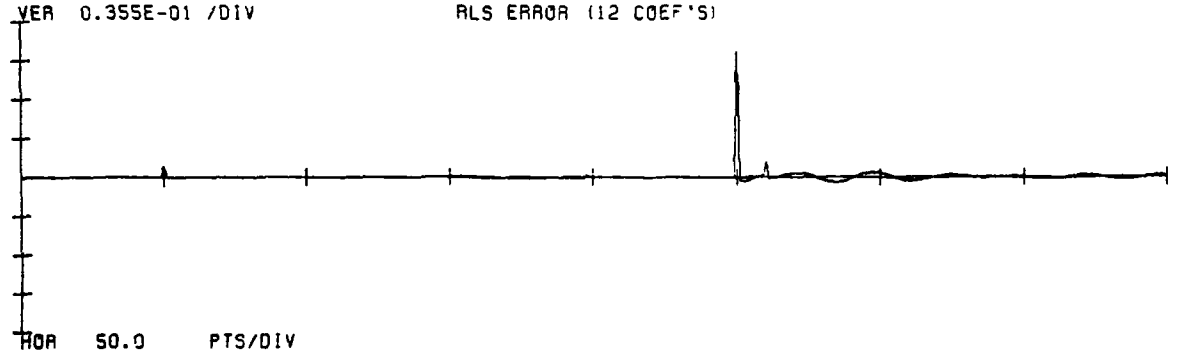
in the data, the coefficient change signal will contain an exponentially decaying pulse whose initial height is equal to the distance between the old and new values of $\mathbf{c}[k]$ and whose decay time is set by the low pass filters used to generate $\tilde{\mathbf{c}}$. In effect the motion of the predictor is being high-passed filtered to create the coefficient change signal. One drawback of this scheme is that slow changes in the predictor will be reduced in amplitude due to the high-passed nature of the coefficient change signal.

The RLS error and the coefficient change signal were use to perform the event compression in all remaining figures. Figure 3.9 shows how they behave on the signal burst1. In this case the 50% decay time of the filters used to generate the change signal was 4 points. We empirically found p/3 (where p is the order of the predictor) to be an effective choice for this parameter. Much shorter decay times led to multiple peaks in the change signal at each event and longer times reduced the resolvability of closely spaced events.

3.3 EXPERIMENTAL RESULTS

The last section presented the behavior of the RLS error and coefficient change signals on noiseless data containing all-pole events. In practice noiseless data is rarely available and it is quite possible that preprocessing


VER 0.312 /DIV
BURST1
HOR 50.0 PTS/DIV


VER 0.355E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV

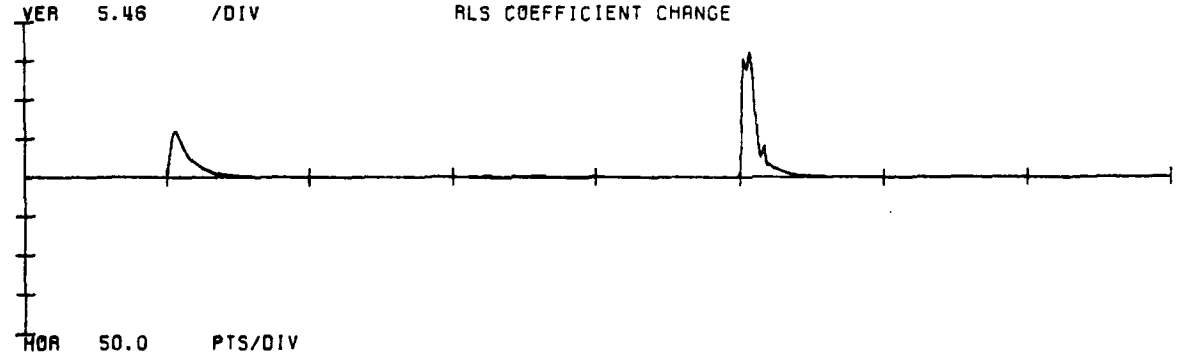

VER 5.46 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.9

(e.g. filtering) could have added zeroes to the events in the data if they were not already present. The following experiments are intended to give the reader some insight as to what to expect from these event compression signals when the input data is not ideal.

3.3.1 Additive Noise

The example of event location given in the last section used noiseless data. Figure 3.10 shows what happens to that example when white gaussian noise is added to the input sequence Burst1 (the standard deviation of this noise is σ=.001 giving the first burst a S/N of 60db). Two important features appear in this figure: the pulse in the coefficient change signal where the algorithm is started (point 0), and the substantial difference between the coefficient response to the second burst (point 150) and the first (point 50). Both the starting transient (in the coefficients) and the large coefficient response to the first event are implied by the structure of the RLS update equations.

Recall that the algorithm is trying to adjust **c** to get the least square error E in the equation

$$E = ||\mathbf{e}||^2 = ||\mathbf{s}-\mathbf{S}\mathbf{c}||^2 \quad . \qquad (3.3)$$

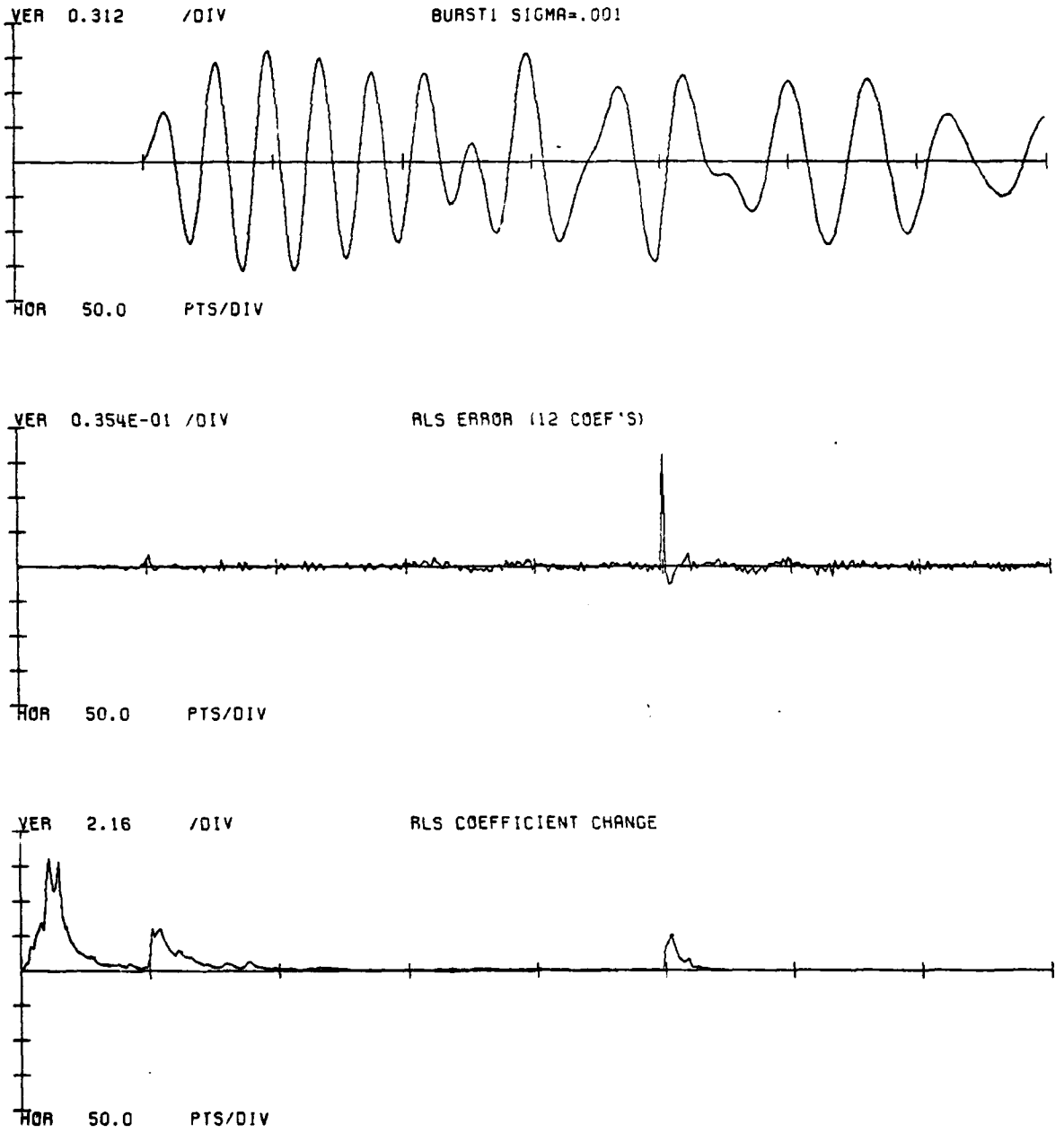

VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.354E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 2.16 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.10

Each new point adds a row to the matrix **S** and new points to the vectors **s** and **e** giving

$$\mathbf{e}' = \begin{Bmatrix} \mathbf{e} \\ \hline e' \end{Bmatrix} = \begin{Bmatrix} \mathbf{s} \\ \hline s' \end{Bmatrix} - \begin{Bmatrix} \mathbf{S} \\ \hline r^T \end{Bmatrix} \mathbf{c}' \tag{3.4}$$

and the new predictor **c'** minimizes $E'=||\mathbf{e}||^2+e'^2$. By separating the components of the new error E' into the contribution from previous points E_p and the contribution from the current point E_c and by introducing the vector **d** to represent the change in the predictor coefficients (i.e. **d=c'-c**) one has the following relations:

$$E_p = ||\mathbf{e}||^2 + ||\mathbf{Sd}||^2 \tag{3.5}$$

$$= E + \mathbf{d}^T\mathbf{Rd}$$

$$E_c = ||s'-\mathbf{r}^T\mathbf{c}'||^2 \tag{3.6}$$

The term $\mathbf{d}^T\mathbf{Rd}$ represents the error "cost" of changing the predictor in terms of poorer prediction of previous points. At each new point the algorithm trades off that cost with the benefits of improved prediction of the current point

(reduction in E_c) that a change might permit. Therefore, if the cost of changing the predictor is low (e.g. the eigenvalues of **R** are small), the predictor will be very responsive to the data and the values of E_c (and consequently the RLS error) will be small. This is the cause of both the starting transient in the coefficient change signal and the sensitivity of the coefficients to the first event.

To illustrate the impact that the starting transient has on the event compressed signals a series of 9 figures (3.11a through 3.13c) was prepared using the Burst1 signal offset to the right 300 points. Each group of three has additive noise at a different level (i.e. σ=.001 , .01 and .1) and within each group the RLS algorithm was started at three different points (point 0, point 200 and point 300).

The starting position of the iteration has a small but noticeable effect on the compressed events. For the RLS error, the sooner the arrival occurs after the starting point of the iteration, the smaller the event will be. Exactly the opposite is true for the coefficient change signal. Equation (3.5) indicates that the longer the interval between the start of the iteration and a given arrival the more linear equations the predictor has to fit and the less it can afford to adjust itself too the new points. Since the data values generating these additional equations are noise, tne equations are

independent and each one puts more of a constraint on the predictor **c** (That would not be true if the data were all zero or came from a noiseless all pole arrival). In effect, the added points desensitize the predictor leading to less predictor change and, consequently, more prediction error. Fortunately, this effect is gradual and does not appear to change the character of the compressed events. Therefore, the exact starting point of the iteration is not crucial as long as the starting transient itself does not obscure the first arrival.

In certain situations the existence of this starting transient may be a problem due to the lack of "eventless" data at the start of some signals. In that case, some means of initializing the RLS iteration to reduce or eliminate the starting transient would have to be found.

START AT 0

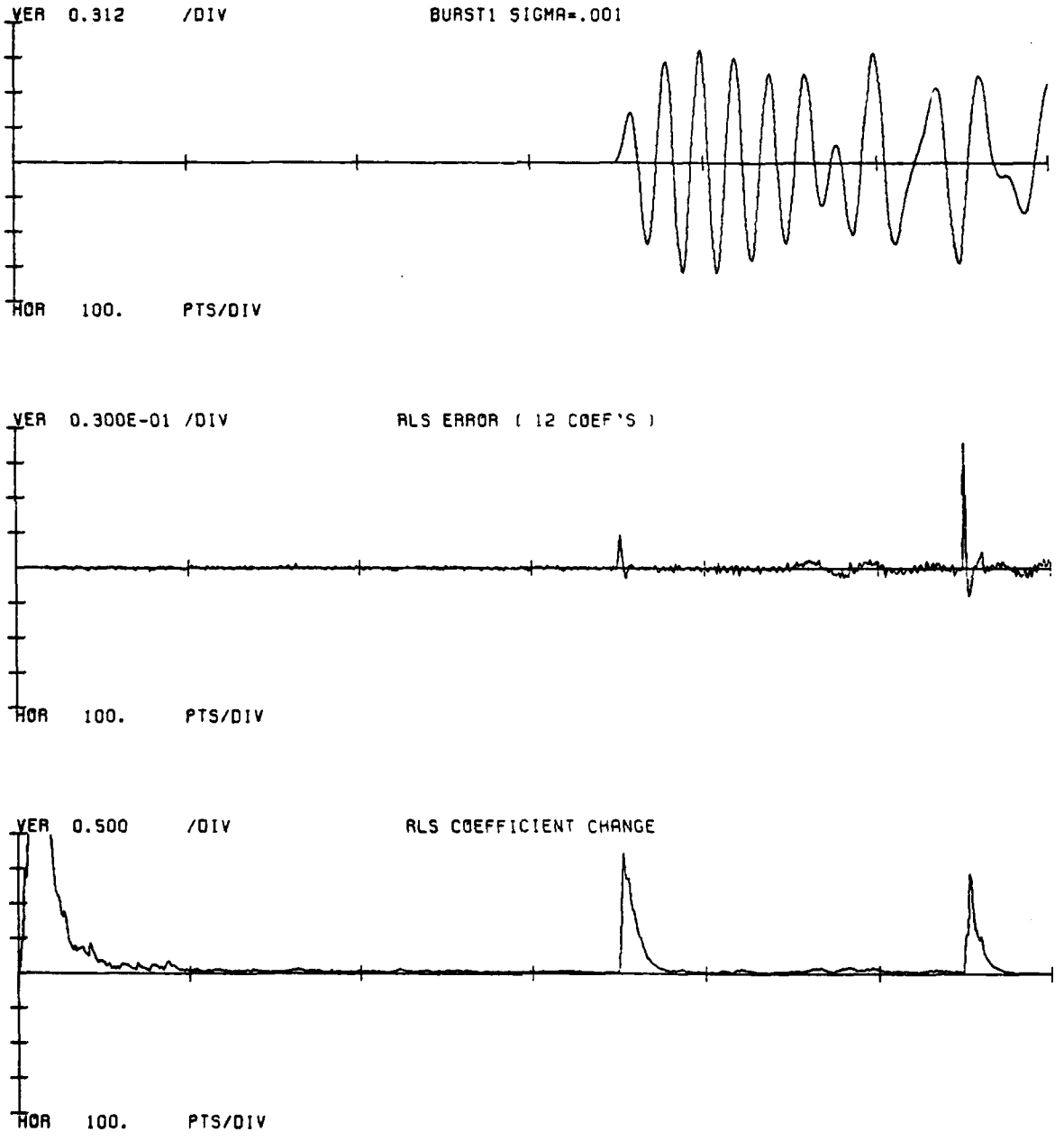


Figure 3.11a

START PT 200

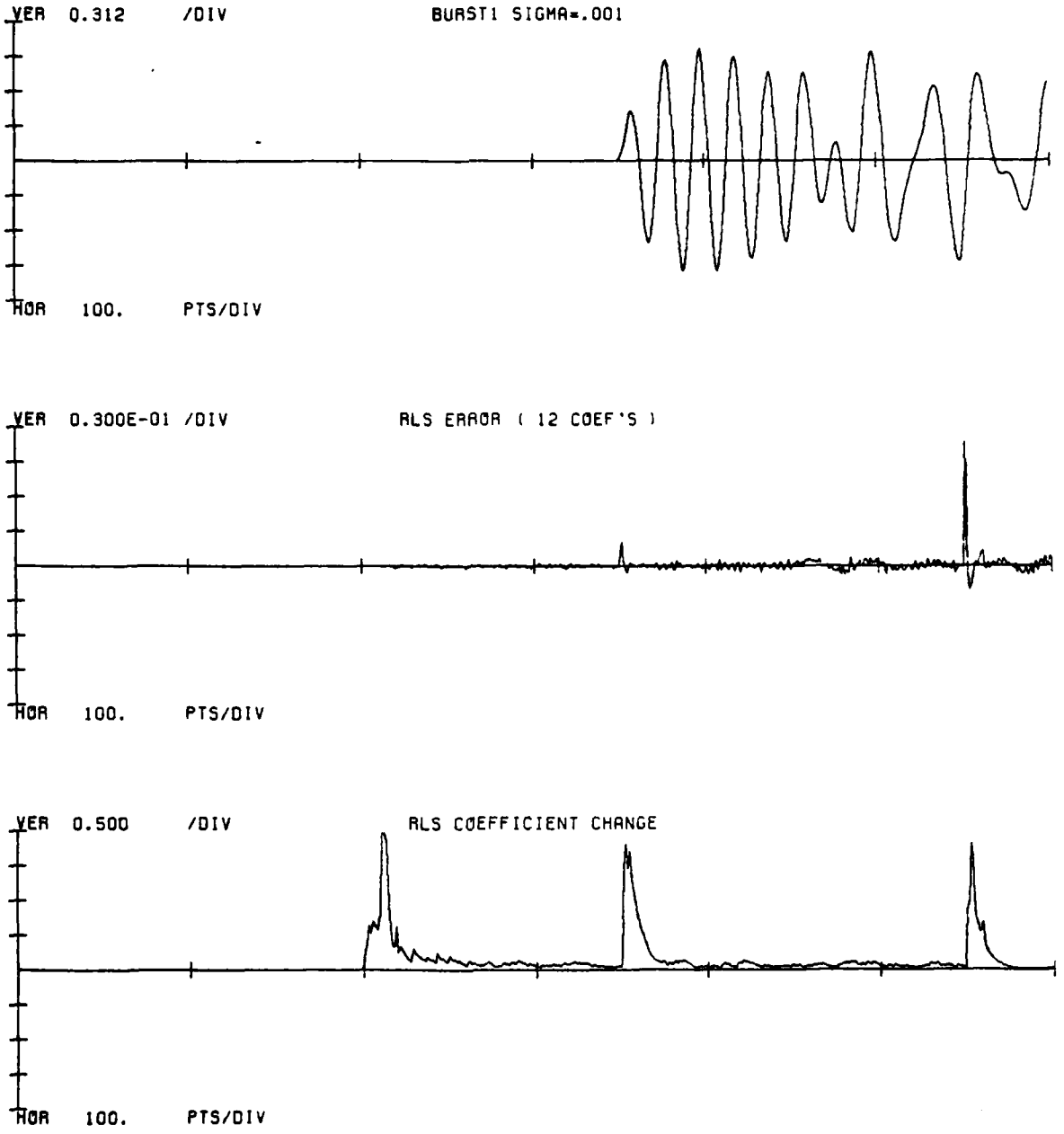


Figure 3.11b

START AT 300

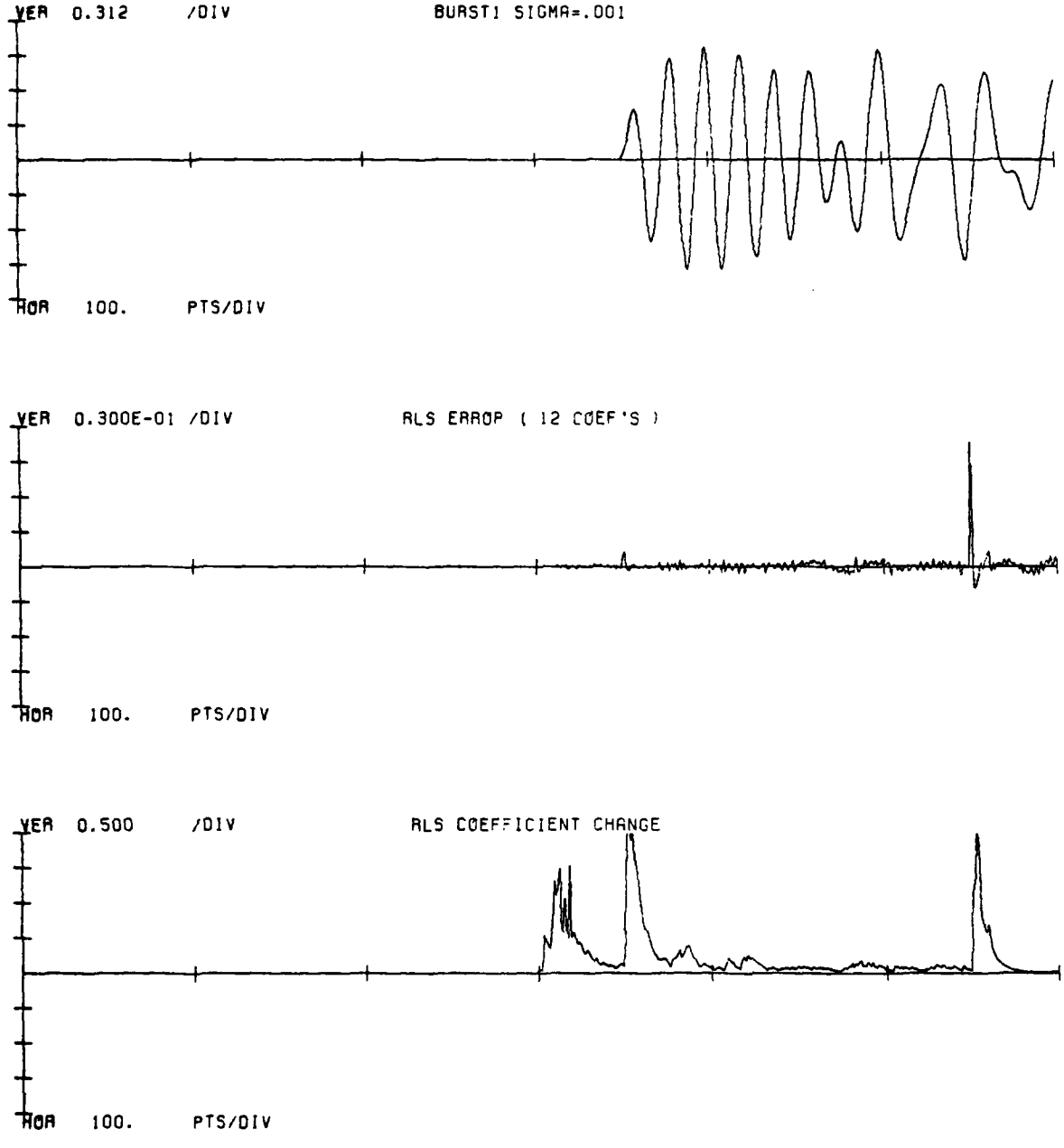


Figure 3.11c

START AT 0

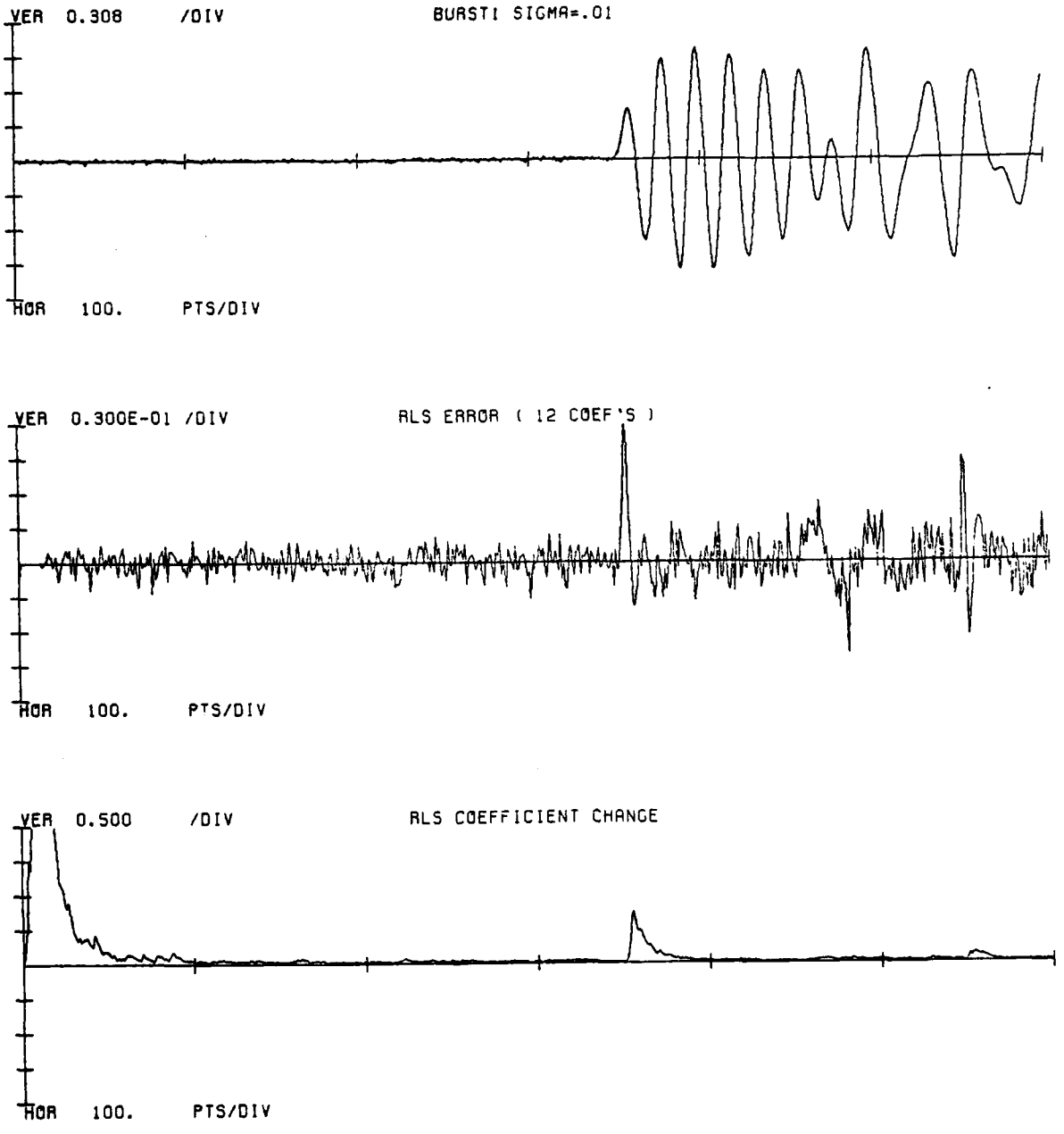


Figure 3.12a

START AT 200

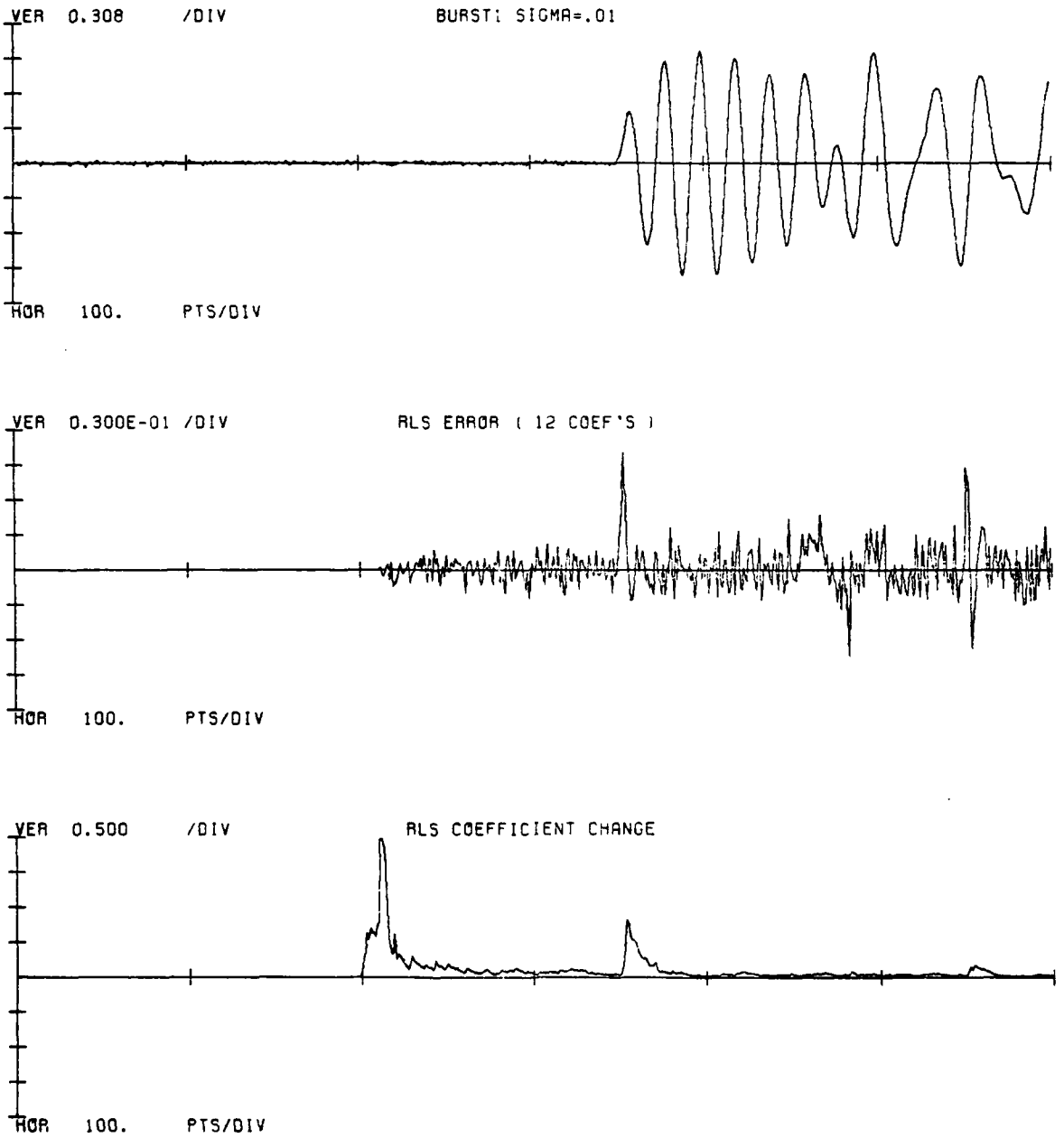


Figure 3.12b

START AT 300

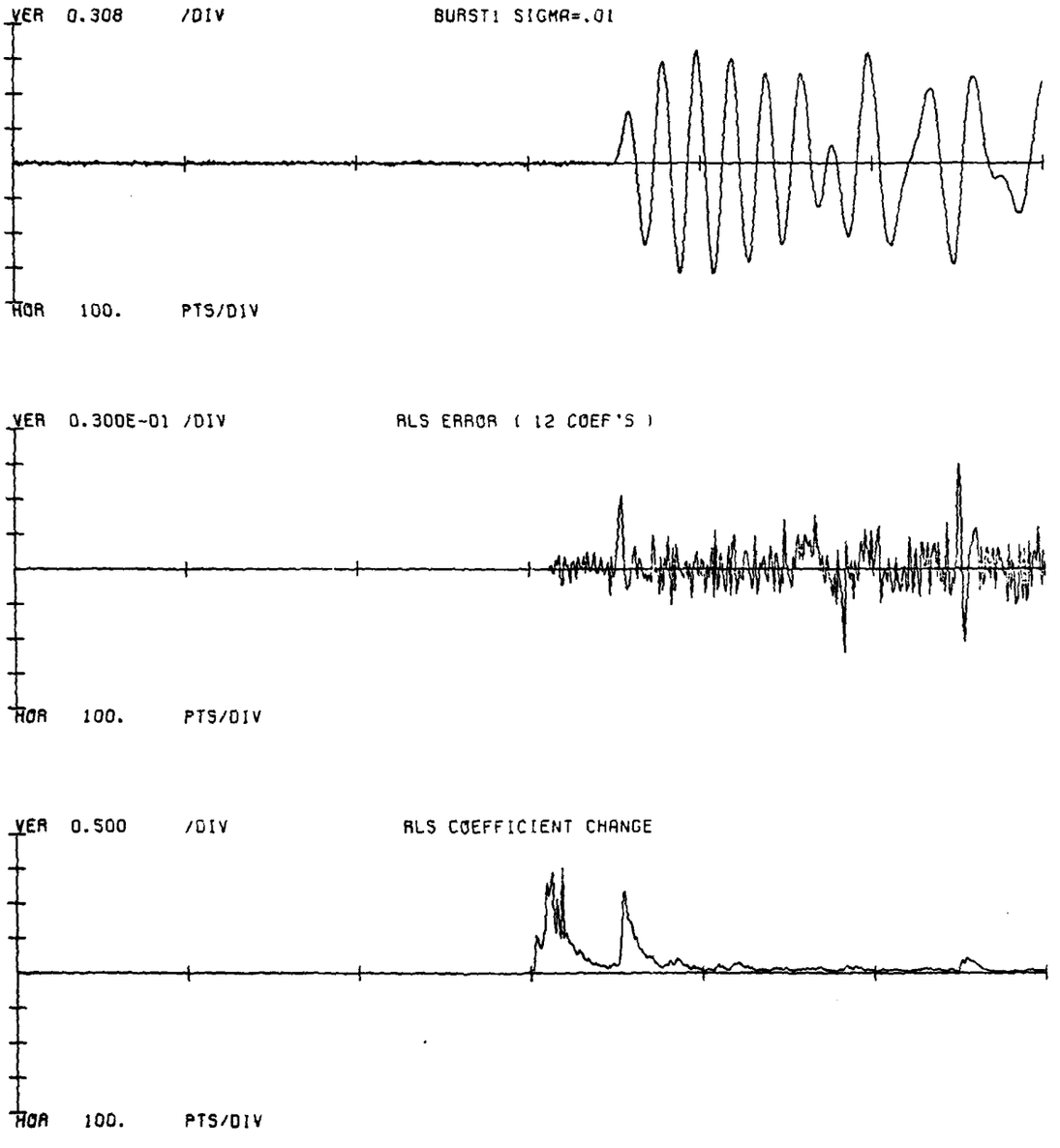


Figure 3.12c

START AT 0

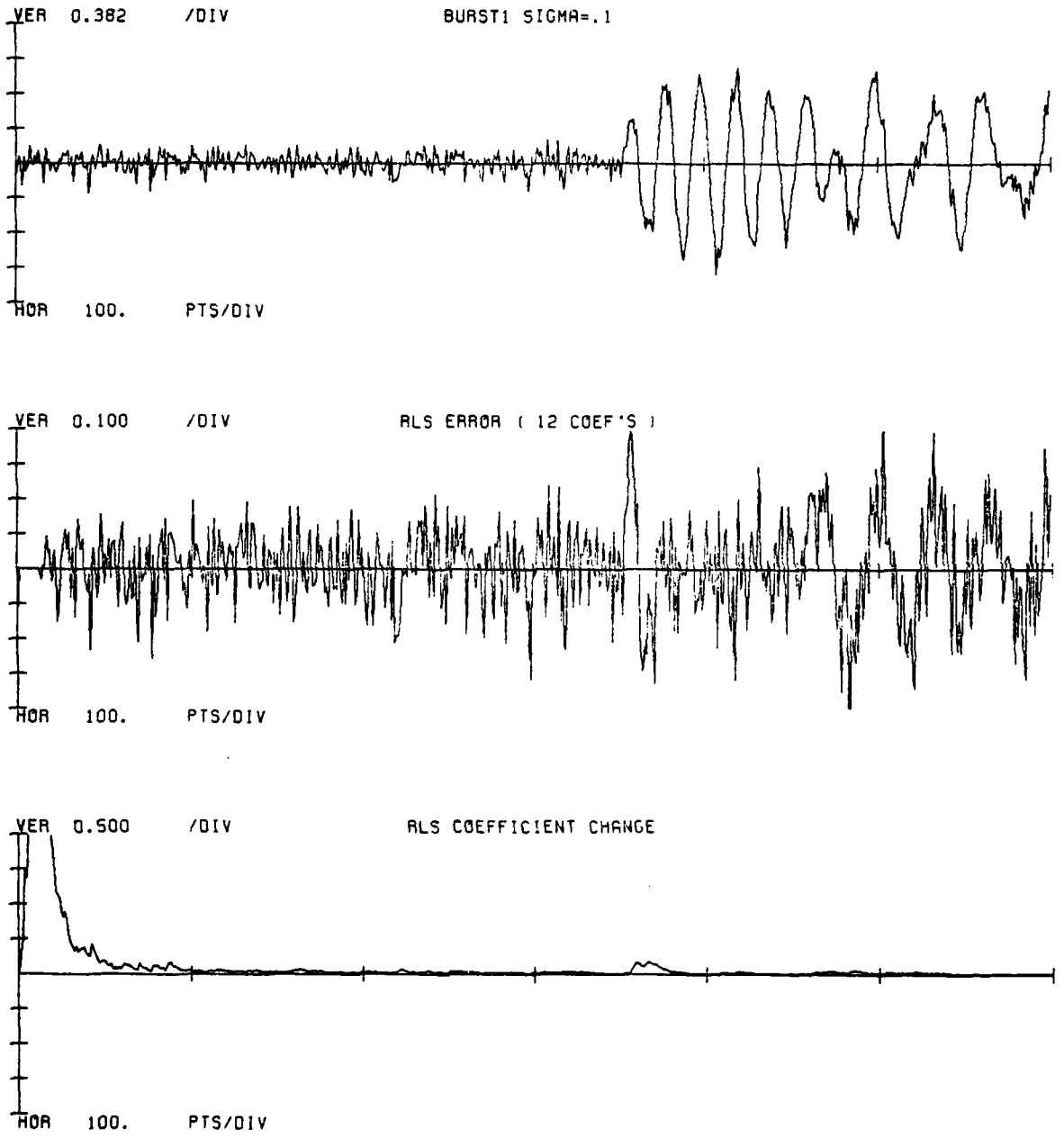


Figure 3.13a

START AT 200

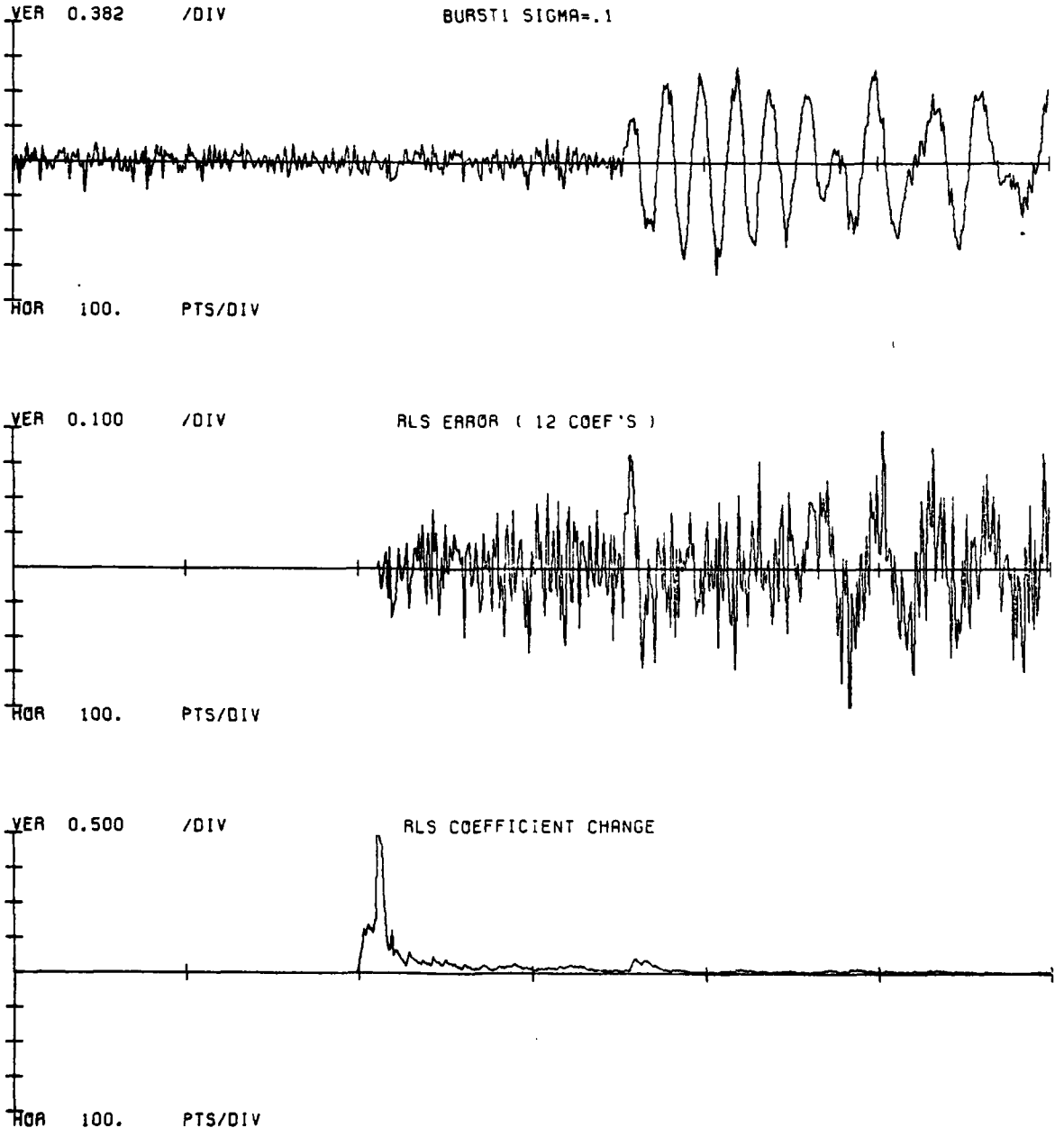


Figure 3.13b

START AT 300

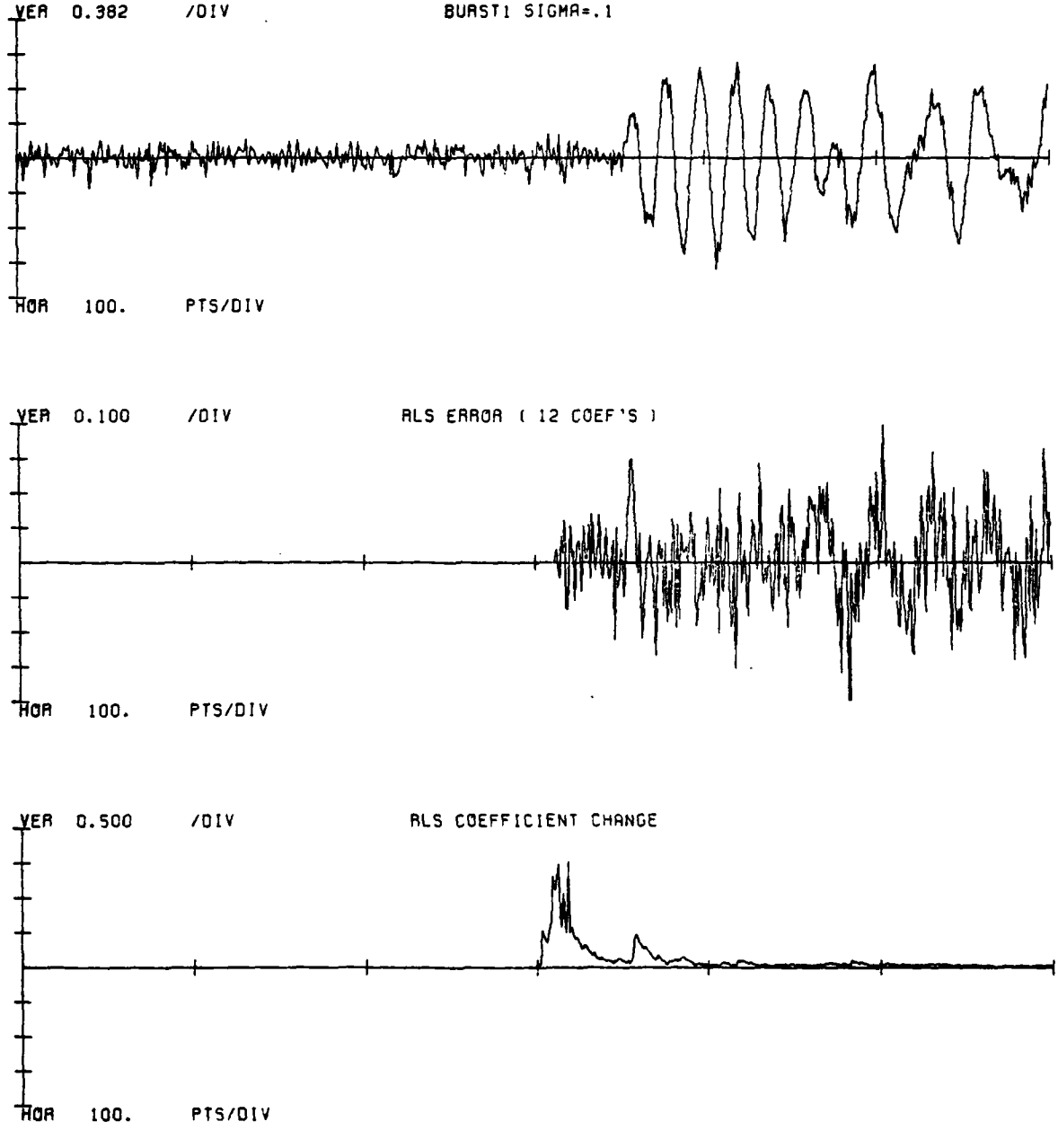


Figure 3.13c

To eliminate the starting transient from all the following event location examples, the predictor was started 500 points to the left of the visible data and consequently the starting transients do not appear. This has no effect other than slightly changing the sizes of the events in the compressed signals.

The noise itself has little effect on the coefficient change signal at a level of -60db (figs. 3.11), but at -40db (figs. 3.12) the size of the third event in the change signal is severely reduced and at -20db (figs. 3.13) only the first event is visible (note the change in scale factor over those three examples). As was noted previously, increasing the noise level in the region before an event reduces the sensitivity of the coefficients to that event leading to a smaller coefficient change.

The RLS error degrades in a different fashion; the most prominent feature being the apparently magnified noise in the RLS error signal. If one views the problem from a filtering standpoint the equation

$$\left\langle \begin{array}{c|c} & \\ \mathbf{s} & \mathbf{S} \\ & \end{array} \right\rangle \begin{bmatrix} 1 \\ \hline -\mathbf{c} \end{bmatrix} = \mathbf{e} \qquad (3.7)$$

must be solved to minimize $||\mathbf{e}||^2$ or, equivalently, make $\mathbf{e}$ orthogonal to the rows of

$$\left\{ \begin{array}{c|c} & \\ \mathbf{s} & \mathbf{S} \\ & \end{array} \right\} \tag{3.8}$$

In the frequency domain this corresponds to whitening or flattening the spectrum. Examination of the spectrum of the signal Burst1 (in section 3.1) indicates that the flattening will consist largely of raising the high frequency portion of the spectrum. This high boost leads to the noise in the RLS error signal.

The other apparent effect on the RLS error signal of increasing the noise is the increased size of the compressed events. Here again, because increasing the noise reduces the sensitivity of the predictor (thereby lessening the extent to which it adapts to new points); the error that the predictor makes at each new point is larger. Unfortunately, in practice, this magnification of the events in the RLS error fails to keep pace with the noise as the noise is increased. Therefore, this effect does not appear to be a useful means for enhancing the events in the RLS error.

The preceding figures in this section were created

using a pseudo random noise generator which produced exactly the same noise pattern on every graph. Since there may be some question about whether the exact noise pattern has a substantial impact on the appearance of the events in the compressed signals, the last three figures (figs. 3.14a-c) show a fixed noise pattern with the Burst1 signal shifted to three different positions. These figures illustrate that the precise noise pattern has little impact on the characteristics of the events in the location signals.

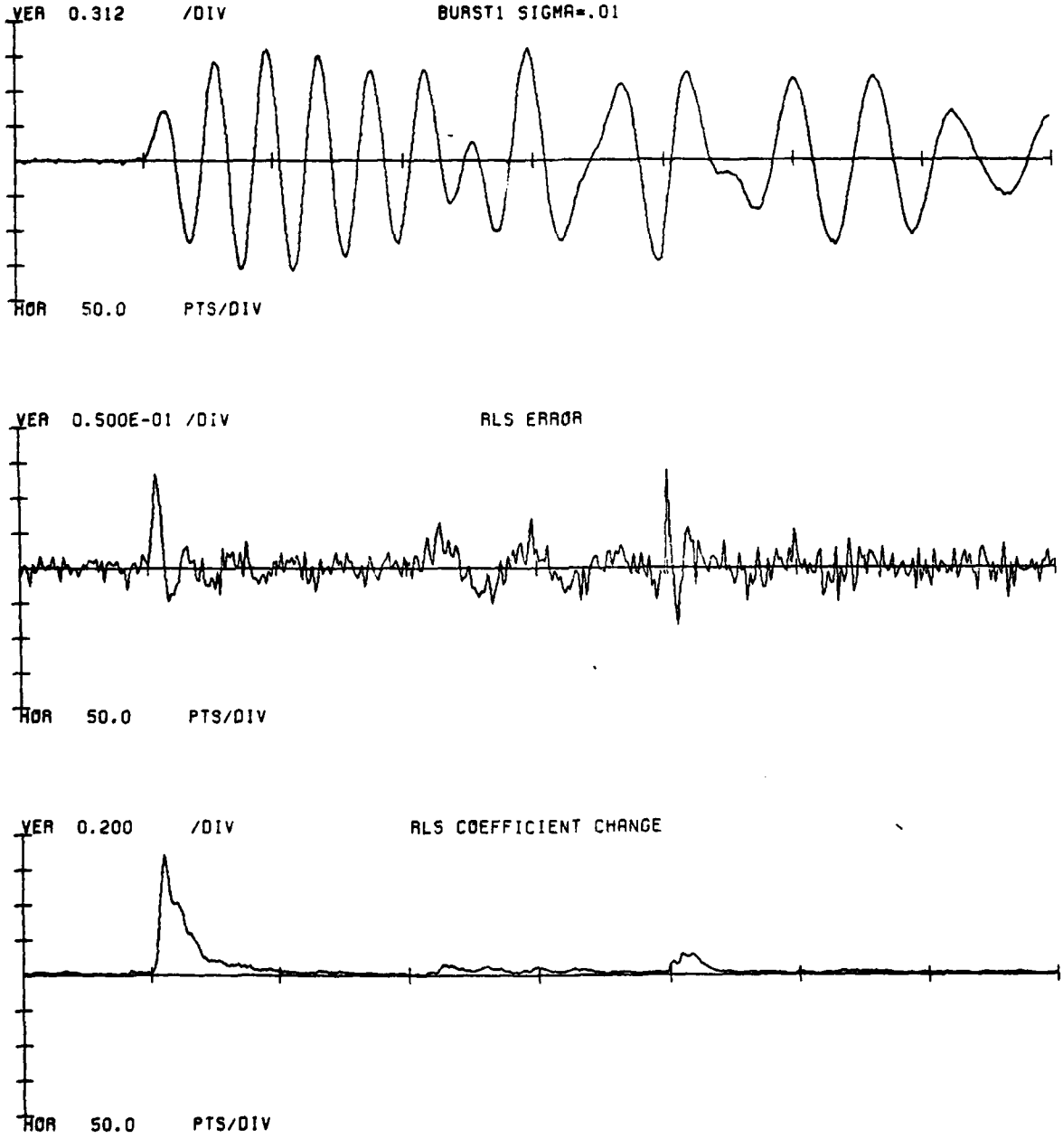

VER 0.312 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.500E-01 /DIV
RLS ERROR
HOR 50.0 PTS/DIV
VER 0.200 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.14a

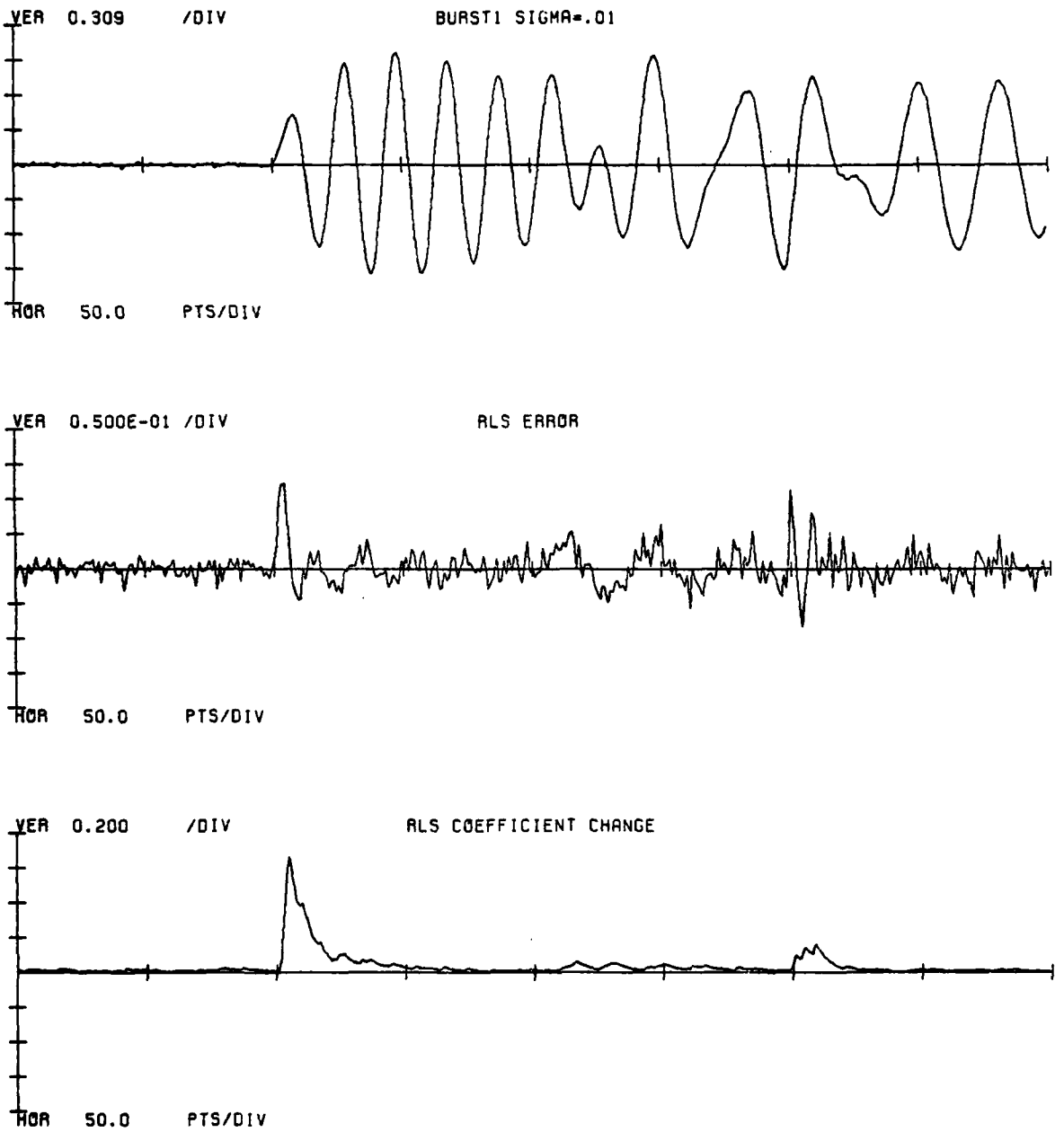

VER 0.309 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.500E-01 /DIV
RLS ERROR
HOR 50.0 PTS/DIV
VER 0.200 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.14b

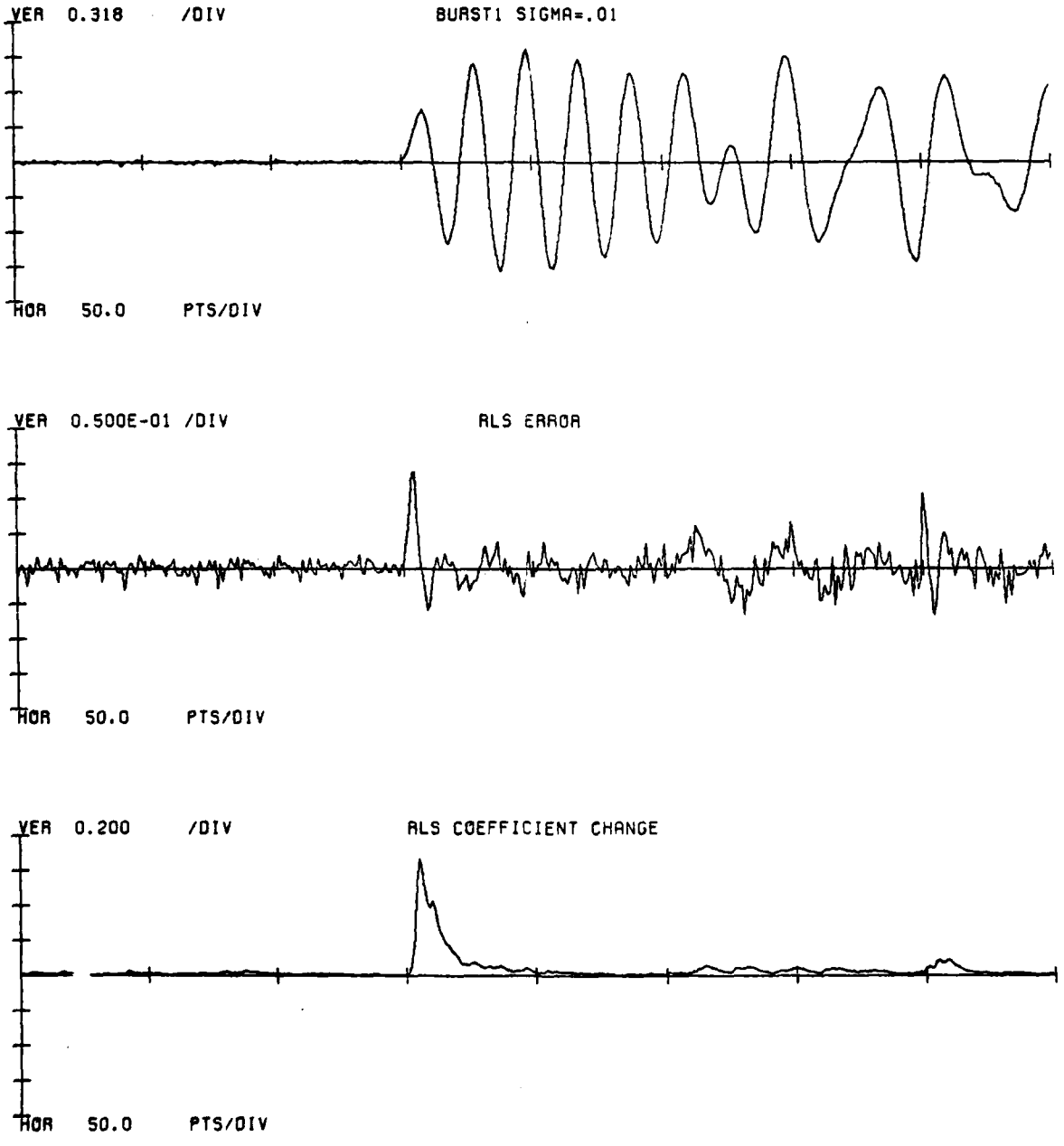

VER 0.318 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.500E-01 /DIV
RLS ERROR
HOR 50.0 PTS/DIV
VER 0.200 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.14c

3.3.1.1 Effects of Predictor Length

Figures 3.15a through 3.17c show four, eight and twelve pole predictors acting on the Burst1 signal at various noise levels. There are two characteristics of event location with RLS visible in this series of figures: first, increasing the noise lengthens the time for the predictor to settle; second, increasing the predictor length (up to a point) decreases the predictor settling time.

The signal covariance matrix **R** (see eq. 3.5) scales in proportion to the noise level. Thus the error energy cost of modifying the predictor at a new point $\mathbf{d}^T\mathbf{R}\mathbf{d}$ increases with the noise level. On the other hand the reduction in error energy at the current point (E_c) from better prediction of the event is independent of the noise level. Therefore, the predictor adapts less to the events in the data as the noise increases and consequently the compressed events in the RLS error and coefficient change signals have longer duration as the noise level increases.

The reason that the predictor settles faster when 8 or 12 coefficients are used rather than 4 is probably because the events in the data contain four poles. Once the second event occurs, a 4 pole model is inadequate. Note that there is very little change in the predictor settling time between the 8 and

12 coefficient predictors when the noise level is the same.[1]

1. In all the experiments which were run, increasing the predictor length beyond 12 coefficients did not improve the location events. However, these signals contained at most three events and the events themselves contained only four poles (at most) each. Situations involving larger numbers of events or very complex events may benefit from larger predictor lengths.

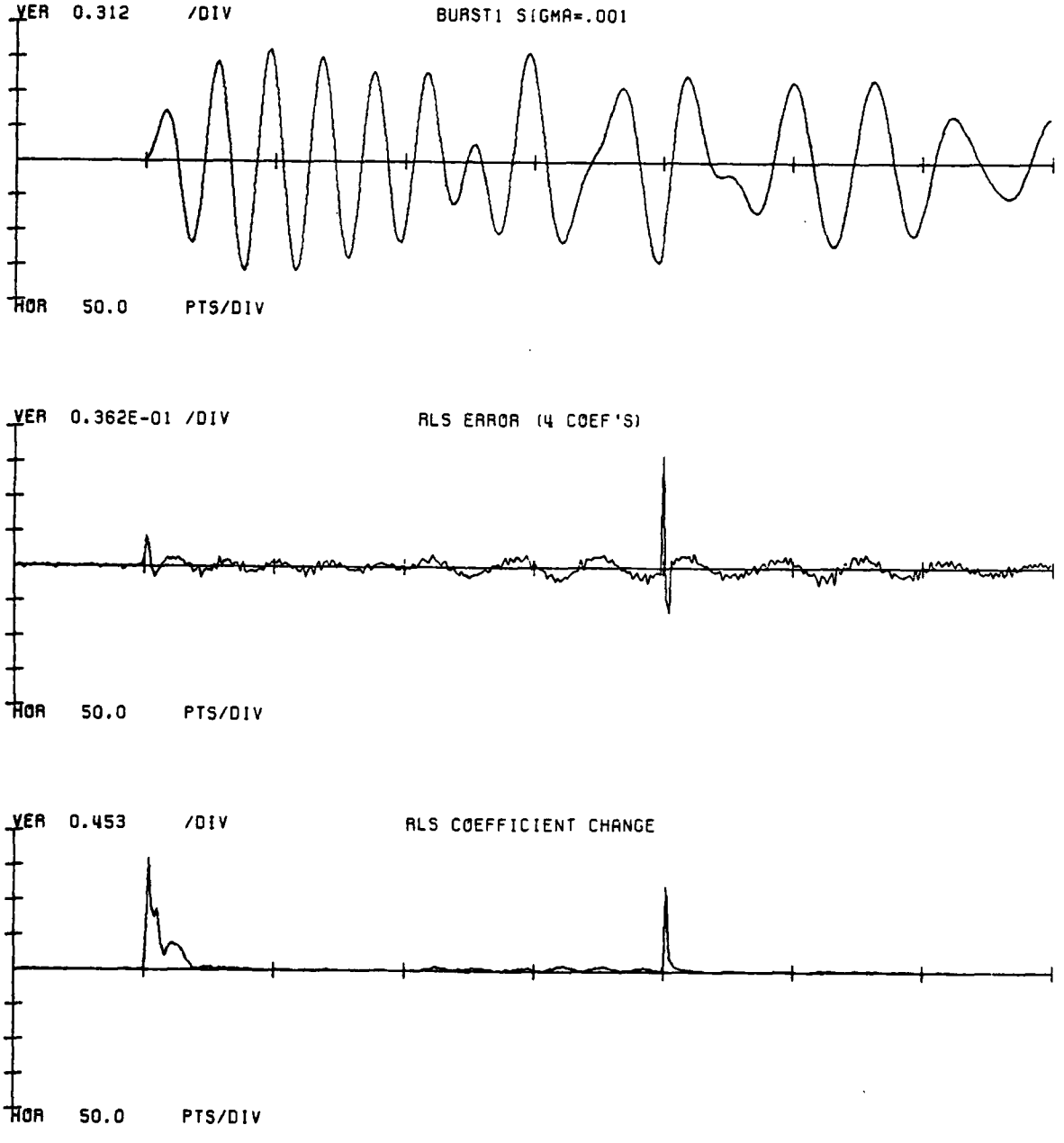

VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.362E-01 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.453 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.15a

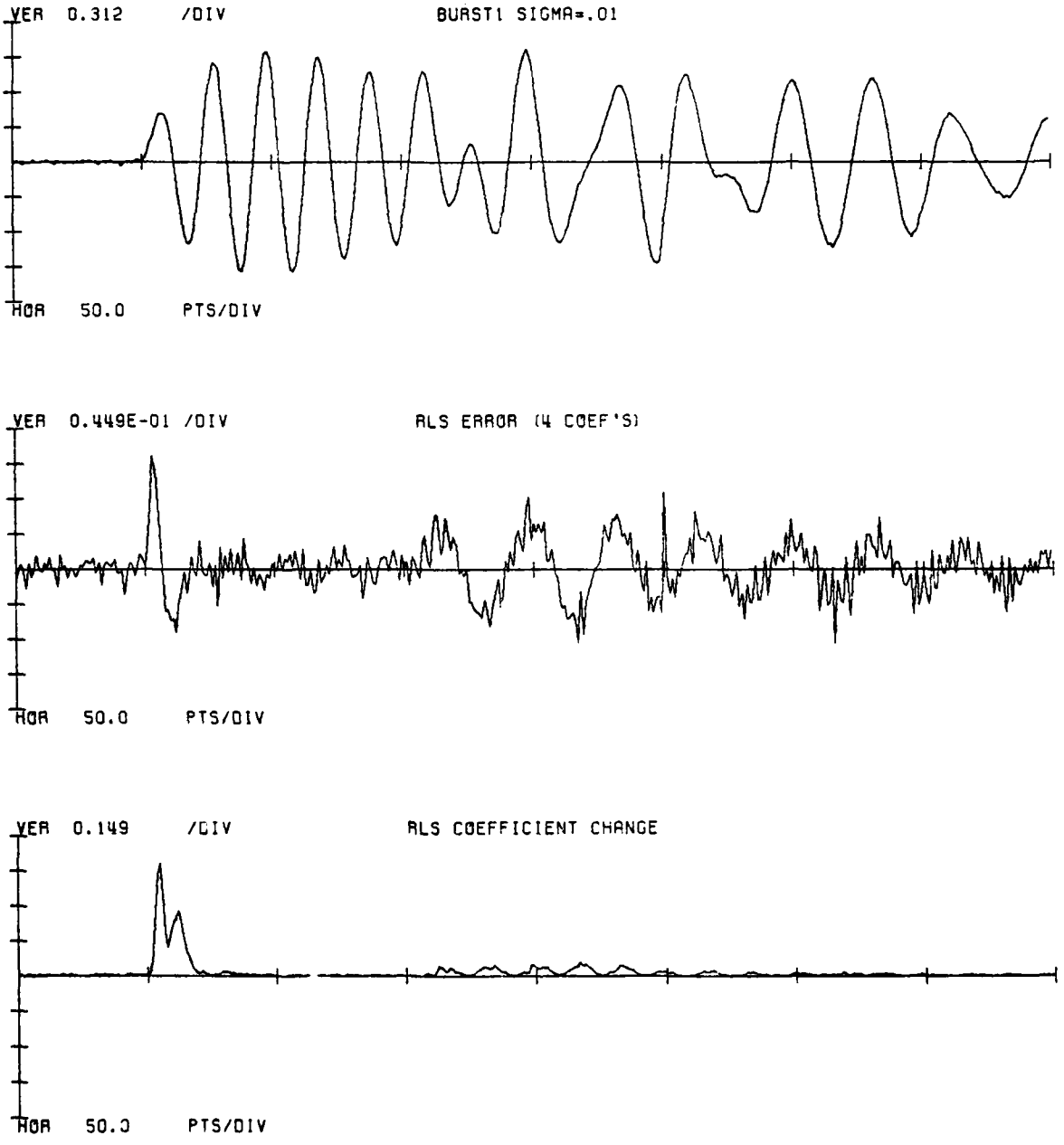

VER 0.312 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.449E-01 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.149 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.15b

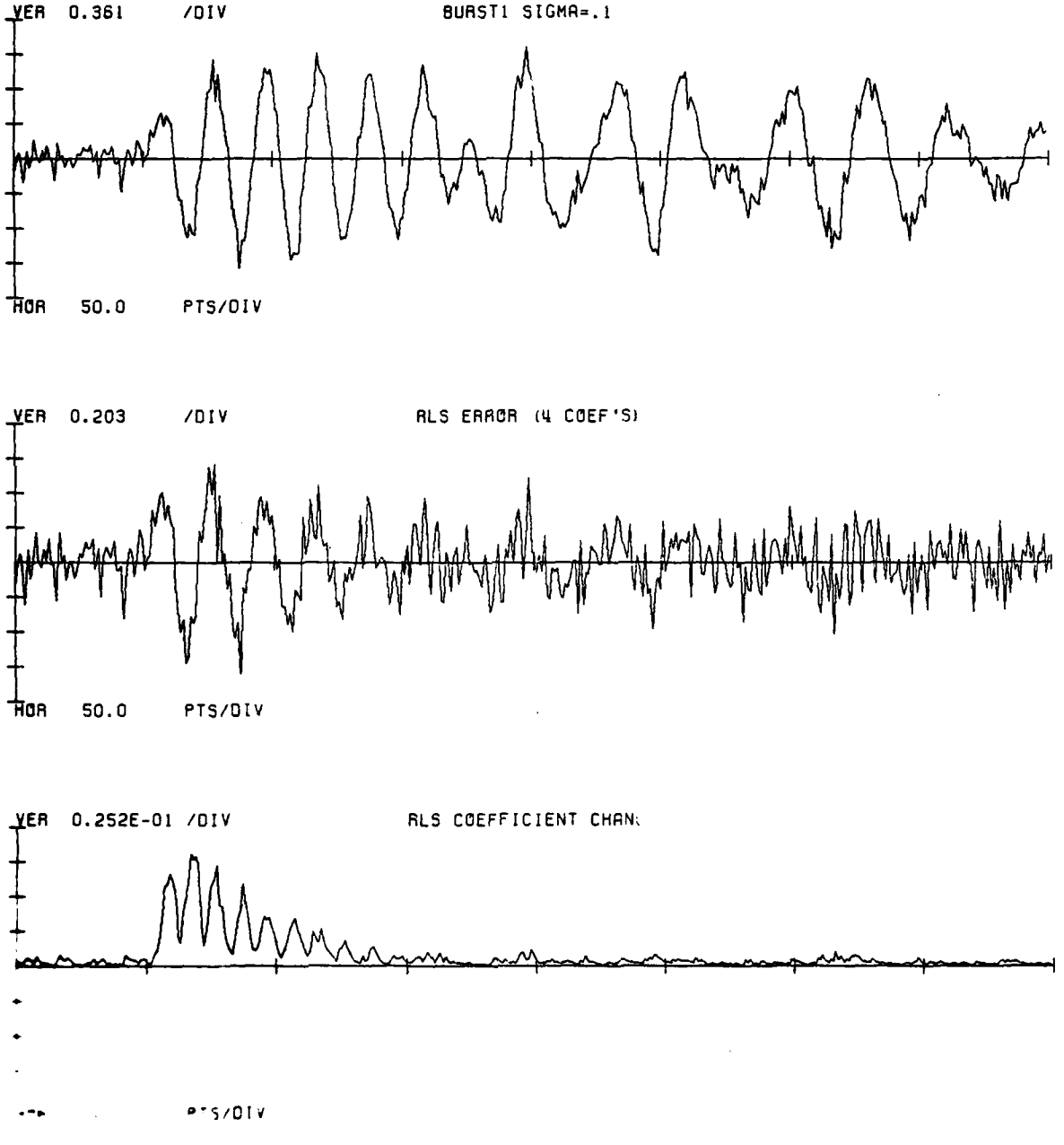

VER 0.361 /DIV
BURST1 SIGMA=.1
HOR 50.0 PTS/DIV
VER 0.203 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.252E-01 /DIV
RLS COEFFICIENT CHAN
PTS/DIV

Figure 3.15c

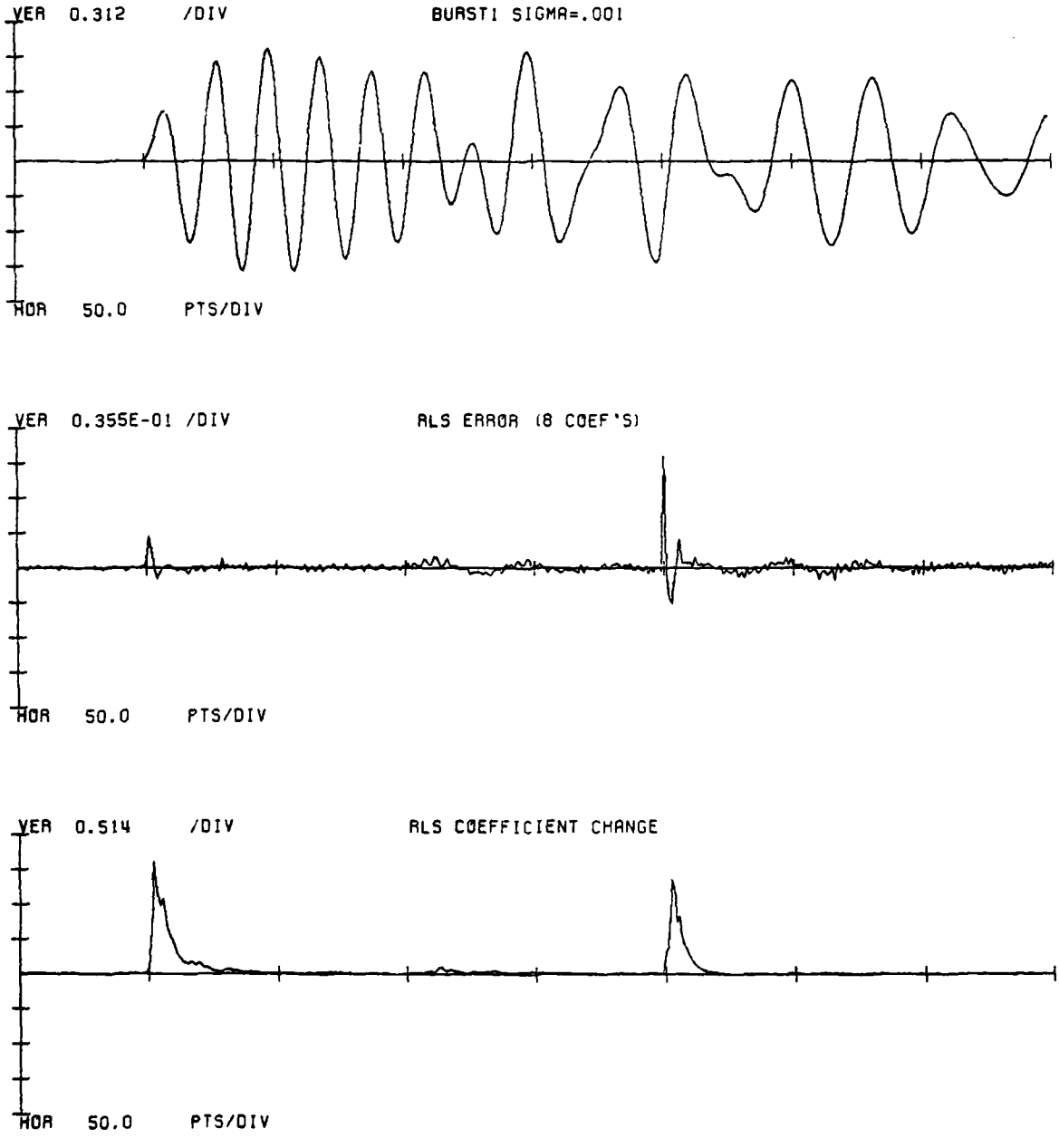

VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.355E-01 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV
VER 0.514 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.16a

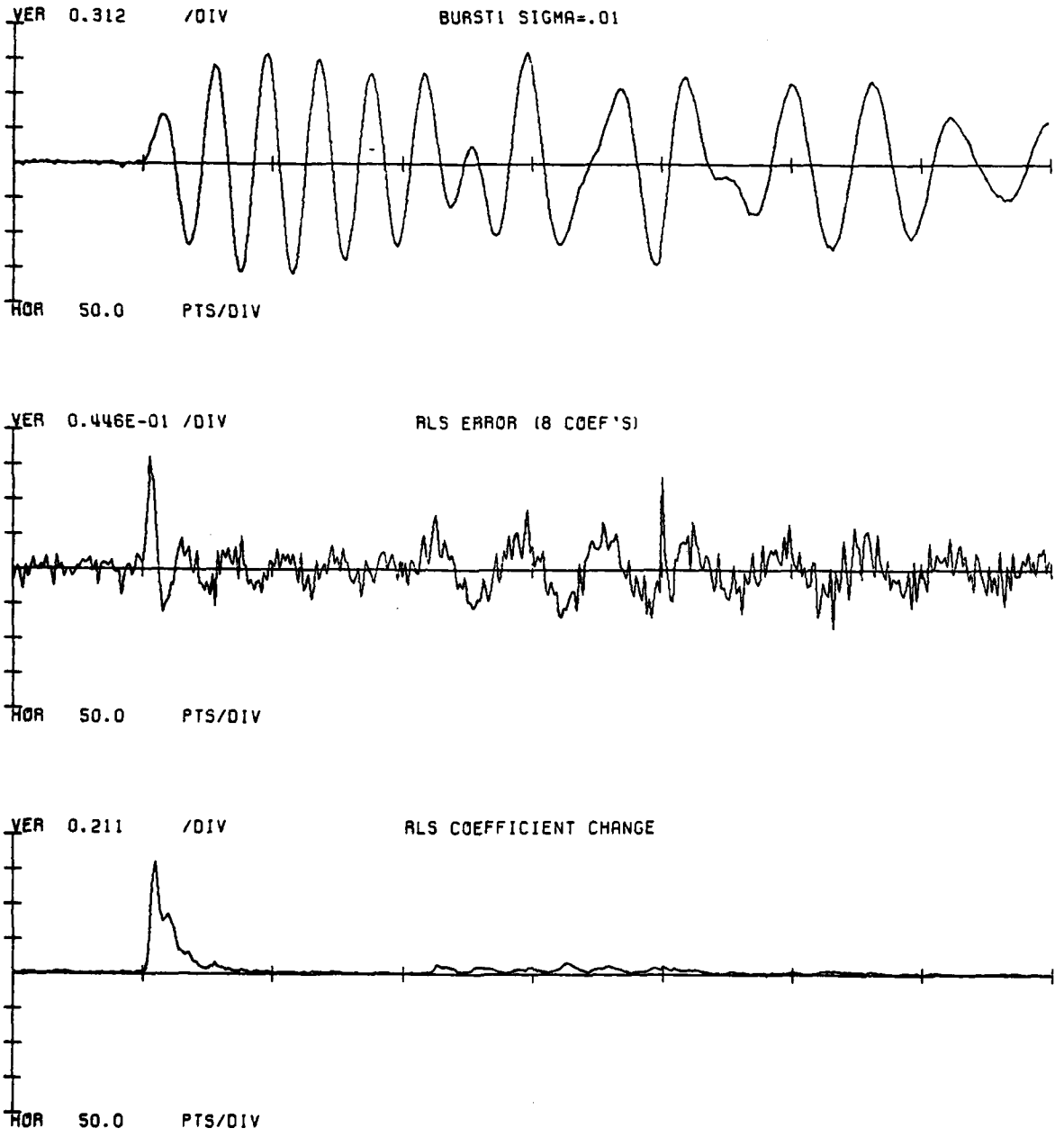

VER 0.312 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.446E-01 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV
VER 0.211 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

figure 3.16b

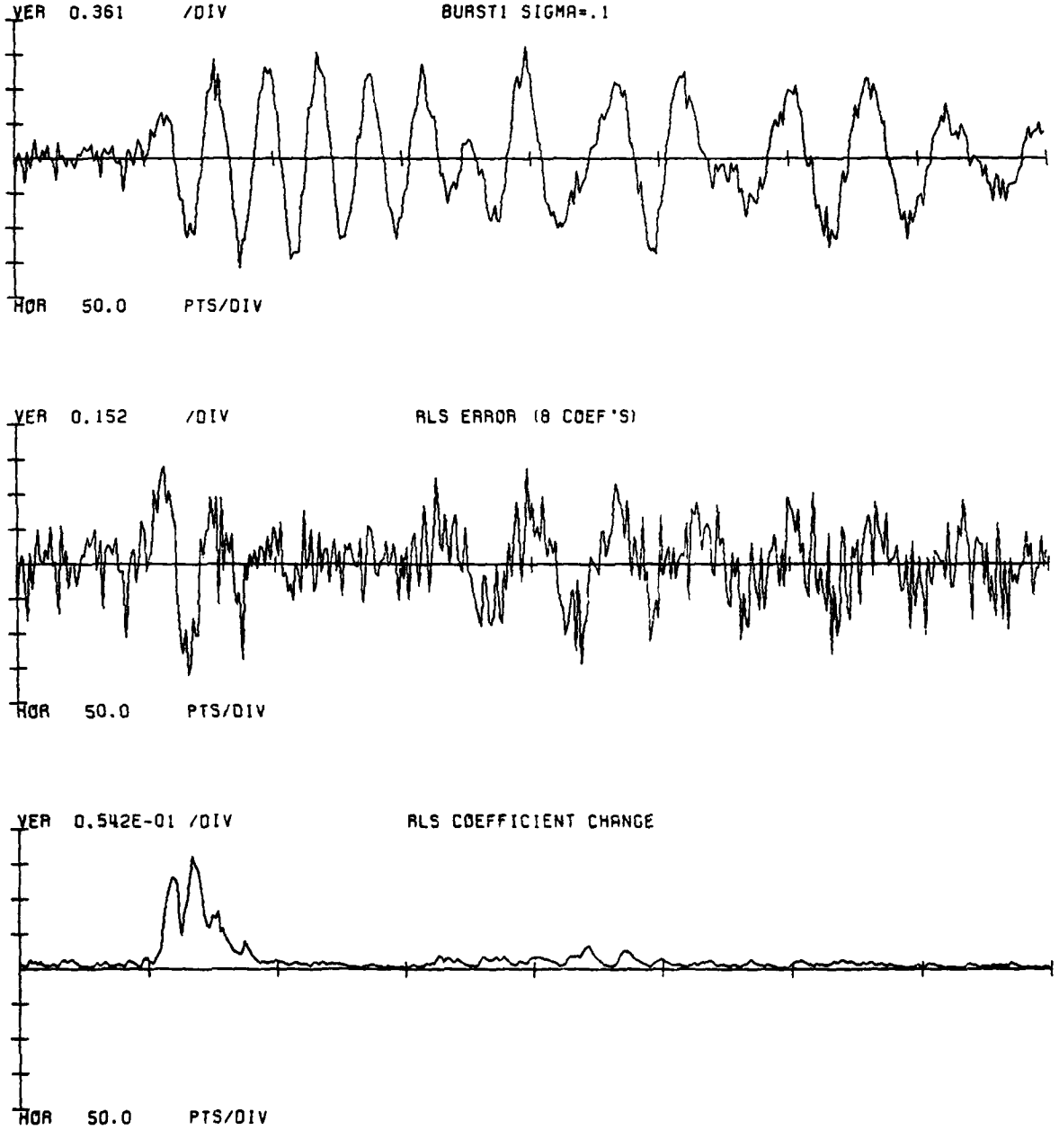

VER 0.361 /DIV
BURST1 SIGMA=.1
HOR 50.0 PTS/DIV
VER 0.152 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV
VER 0.542E-01 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

figure 3.16c

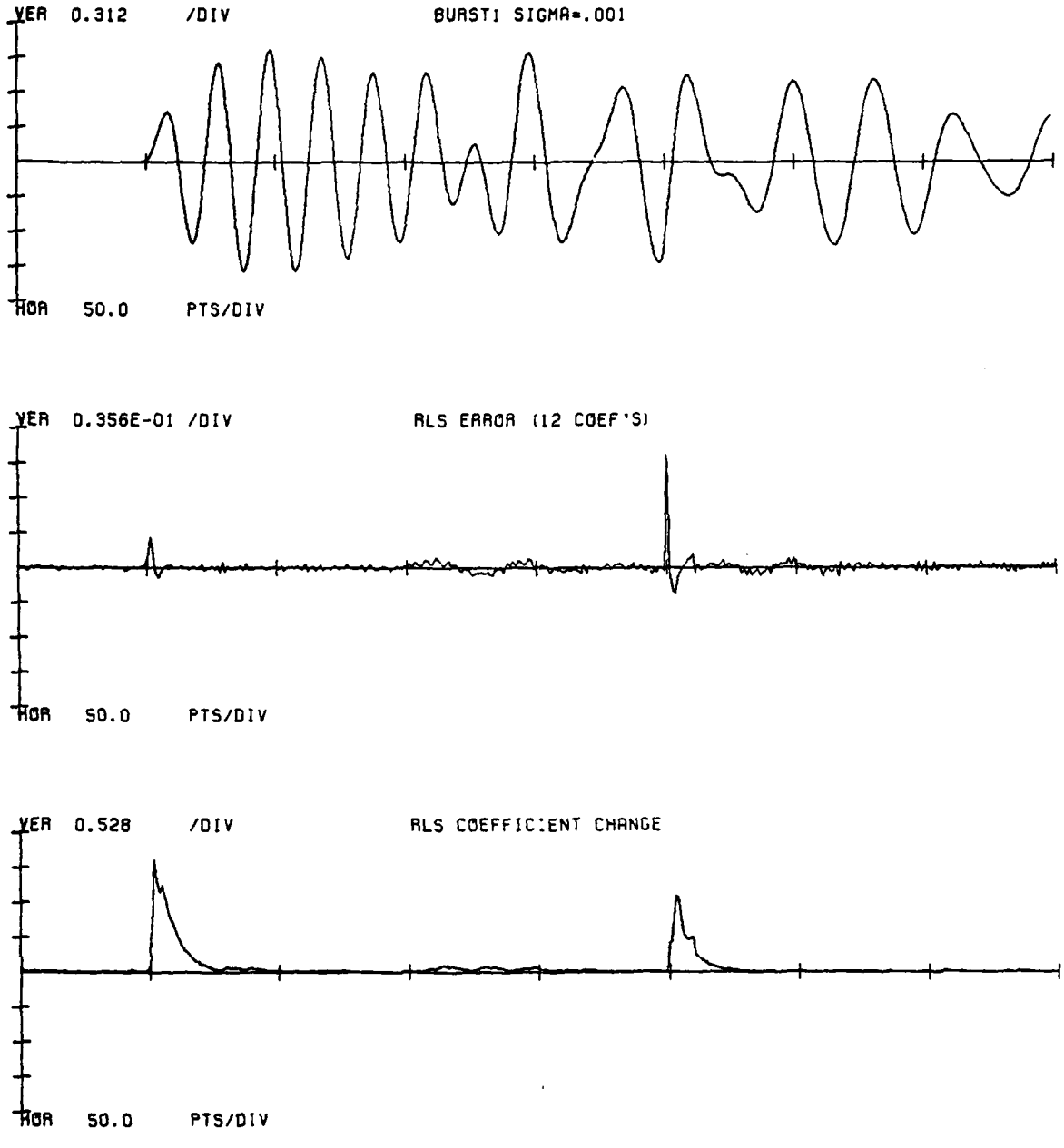

VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.356E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.528 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.17a

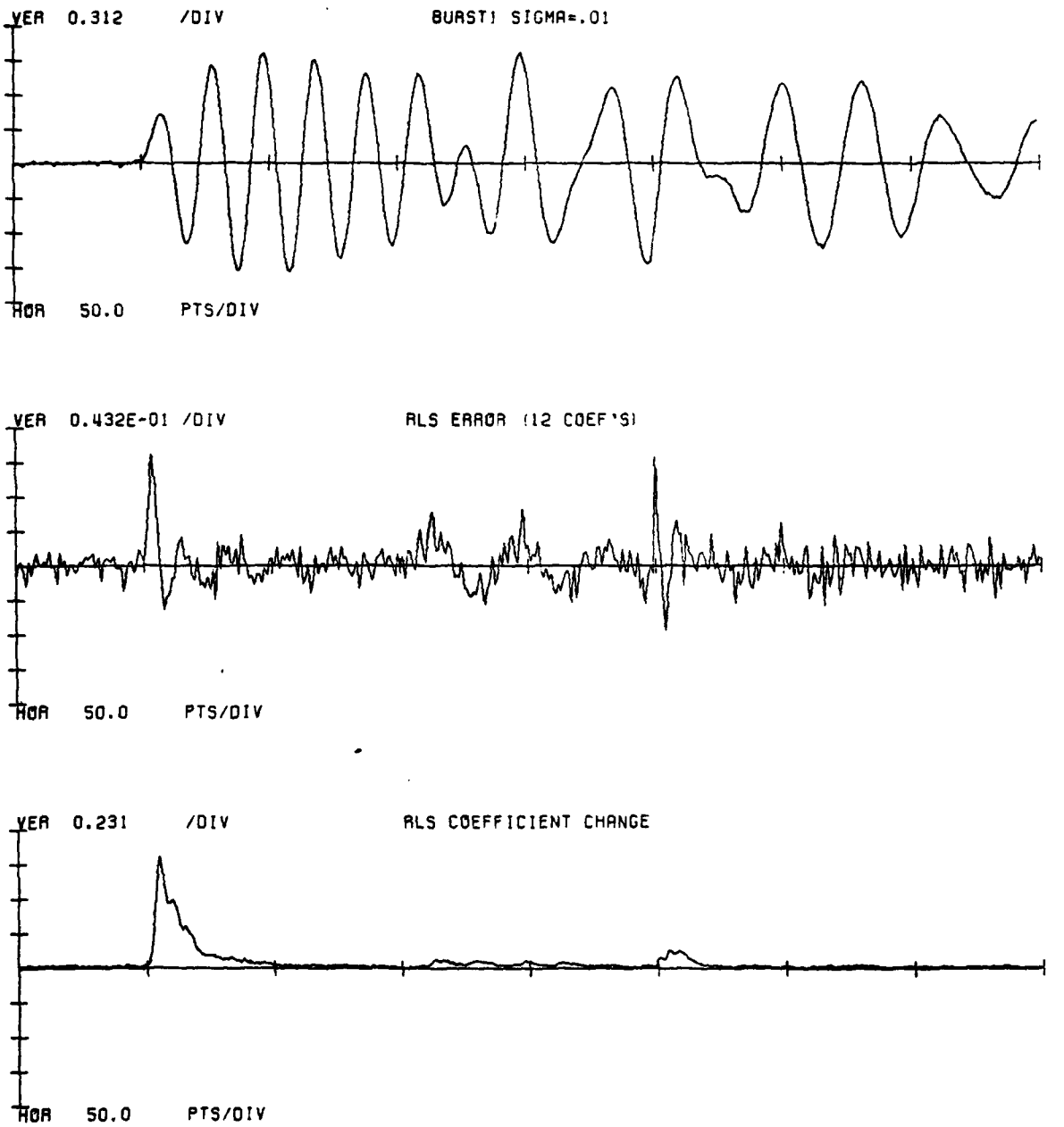

VER 0.312 /DIV
BURST! SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.432E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.231 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.17b

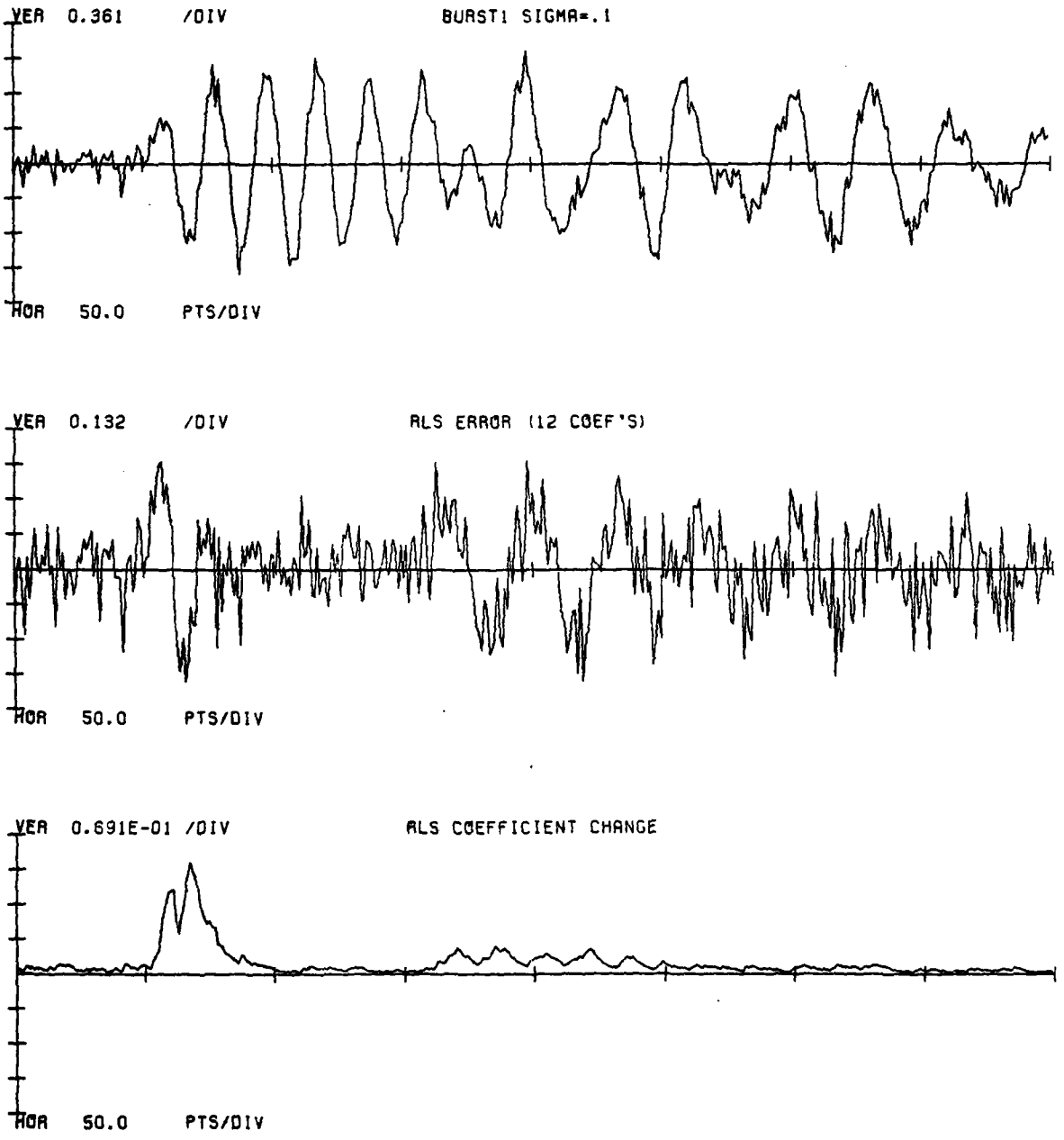

VER 0.361 /DIV
BURST1 SIGMA=.1
HOR 50.0 PTS/DIV
VER 0.132 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.691E-01 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.17c

3.3.2 Interevent Interference

As discussed in the last section, the data preceding an event in part determines how the predictor will react to it. Given a two event situation, the more similar the first event is to the second, the less the prediction error will be at the second event, so the less the predictor will change to adapt to it. Figures 3.18 and 3.19 demonstrate this fact with data consisting of Burstla as the second event and either Burstla or Burstlb as the first.[1] Figure 3.19 (the one with differing events) clearly shows a larger coefficient change and a longer prediction error disturbance at the second event reflecting the larger difference between the two arrivals.

---

1. To make the details of the second event more apparent these graphs have been magnified causing some of the compressed events to go off scale.

BURST1A @ 50 & 150

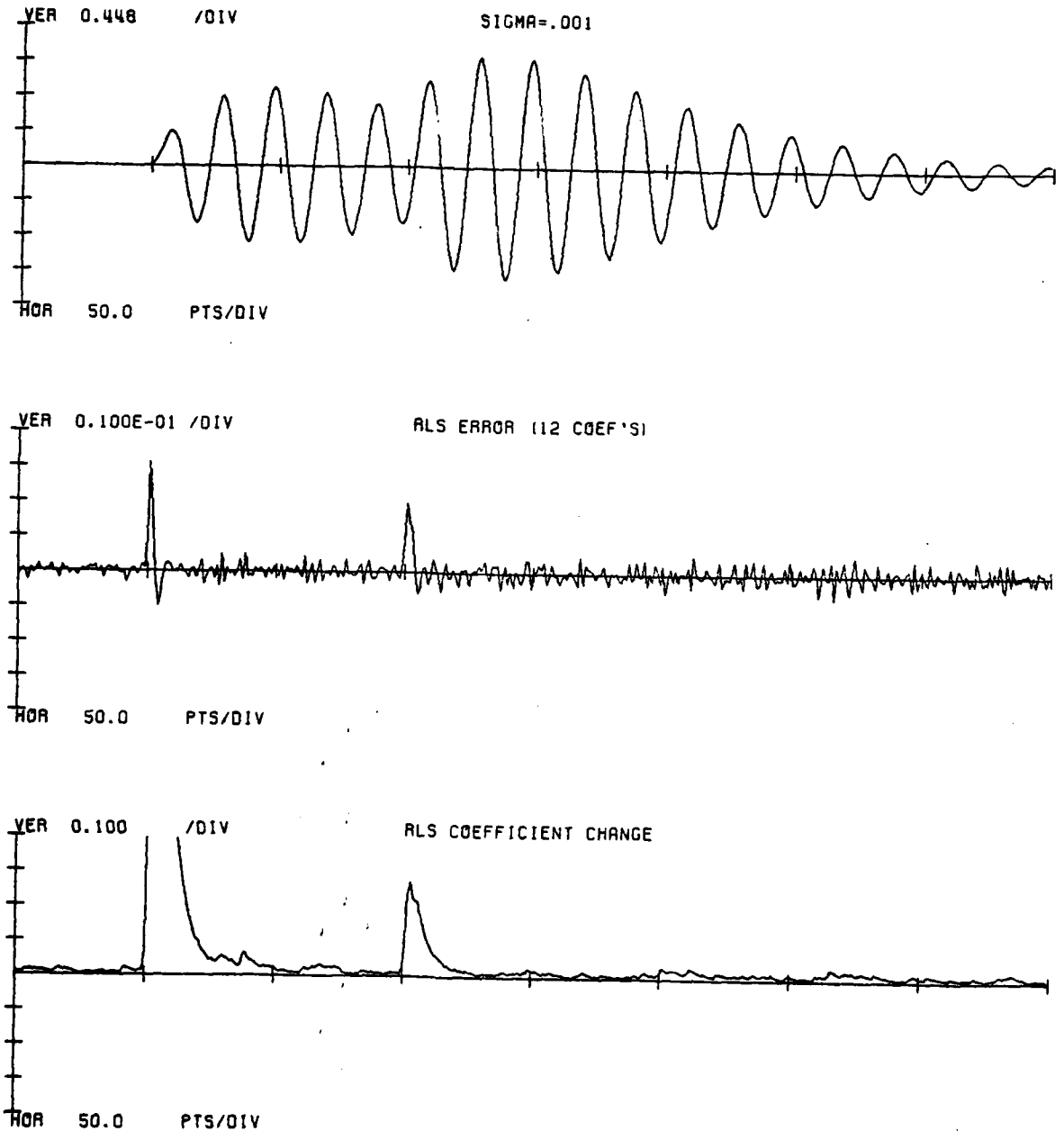


Figure 3.18

BURST1B @ 50 BURST1A @ 150

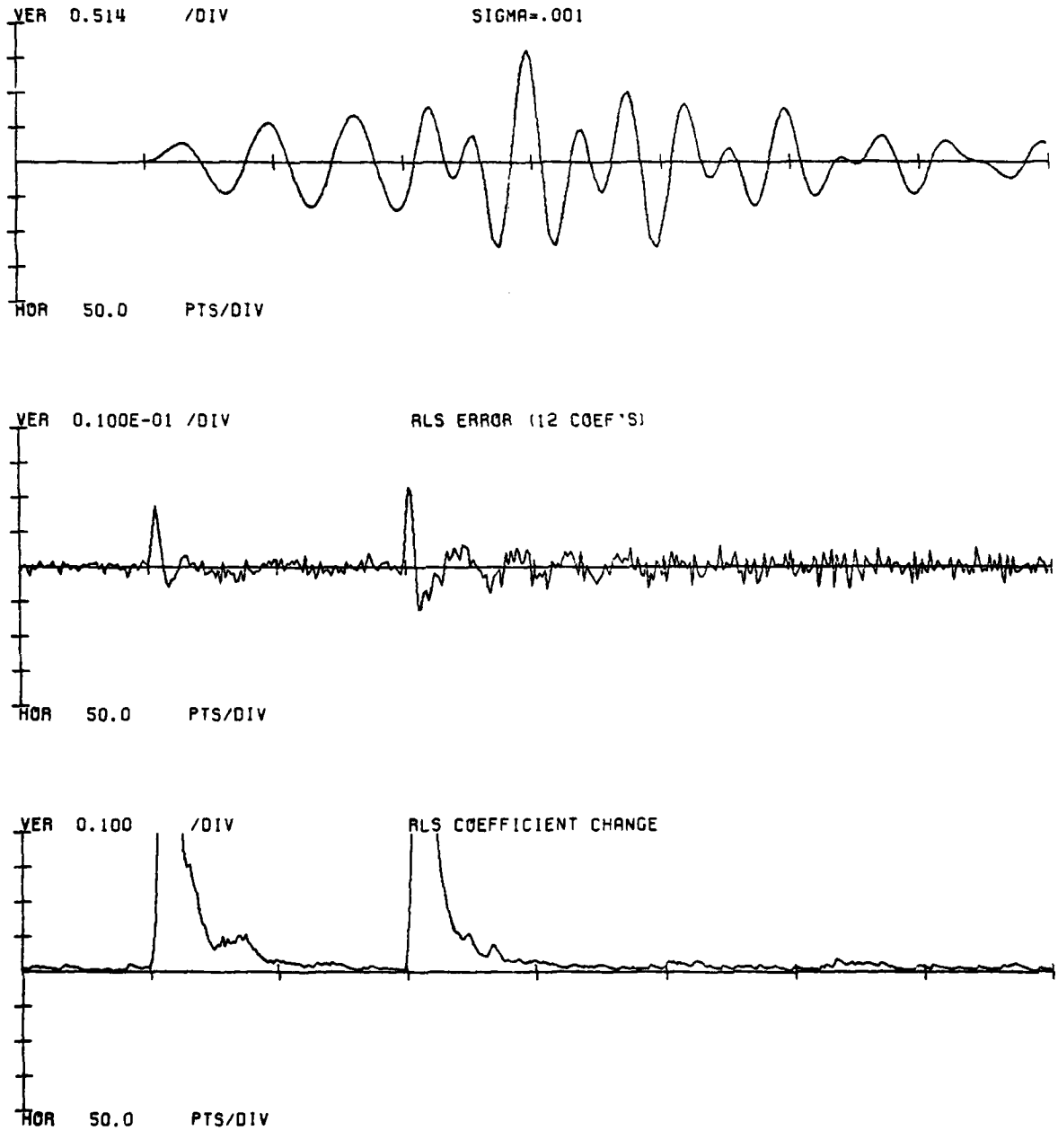


Figure 3.19

The next twelve figures (nos. 3.20a through 3.22d) demonstrate the effect that spacing has on interevent interference. Within each group of four figures (3.20, 3.21 or 3.22) a fixed predictor length (4, 8 or 12 coefficients) was used to compress events in data containing two instances of the Burstla pulse at four different spacings. These figures illustrate how the event compressed signals behave versus event spacing and predictor length.

The compressed events for the four coefficient predictor appear different at all spacings up to 150 points. These differences include changes in size and shape. Examining figure 3.20d reveals a small disturbance in the error sequence after the first event which extends at least 150 points beyond it. The duration of this disturbance appears to determine the zone over which the position of the second compressed event will influence its shape or size. The 8 and 12 coefficient predictors have a much shorter error disturbance, and they generate compressed events which are similar in size and shape for all spacings above 25 points. This led us to believe that the longer the predictor takes to model a given event, as indicated by the time taken for the error disturbance to die away, the further the next event must be to remain unaffected.

This postulation was tested by the following

experiment: signals were generated with Burstlc starting at point 50, Burstlb at point 200, and Burstla at various positions between 100 and 350 points; then event compression was performed. The results are presented in figures 3.23a-h.[1] As can be seen from the figures, the compressed events for the last two bursts only interfere if the predictor does not have time to settle between them. The question of what determines this settling time is a possible topic for future investigation.

---

1. Burstlc was used to desensitize the predictor and it produces a large event simply because it is first. To make the details of the remaining events clearer, the RLS error and coefficient change graphs were magnified causing the first compressed event to go off scale.

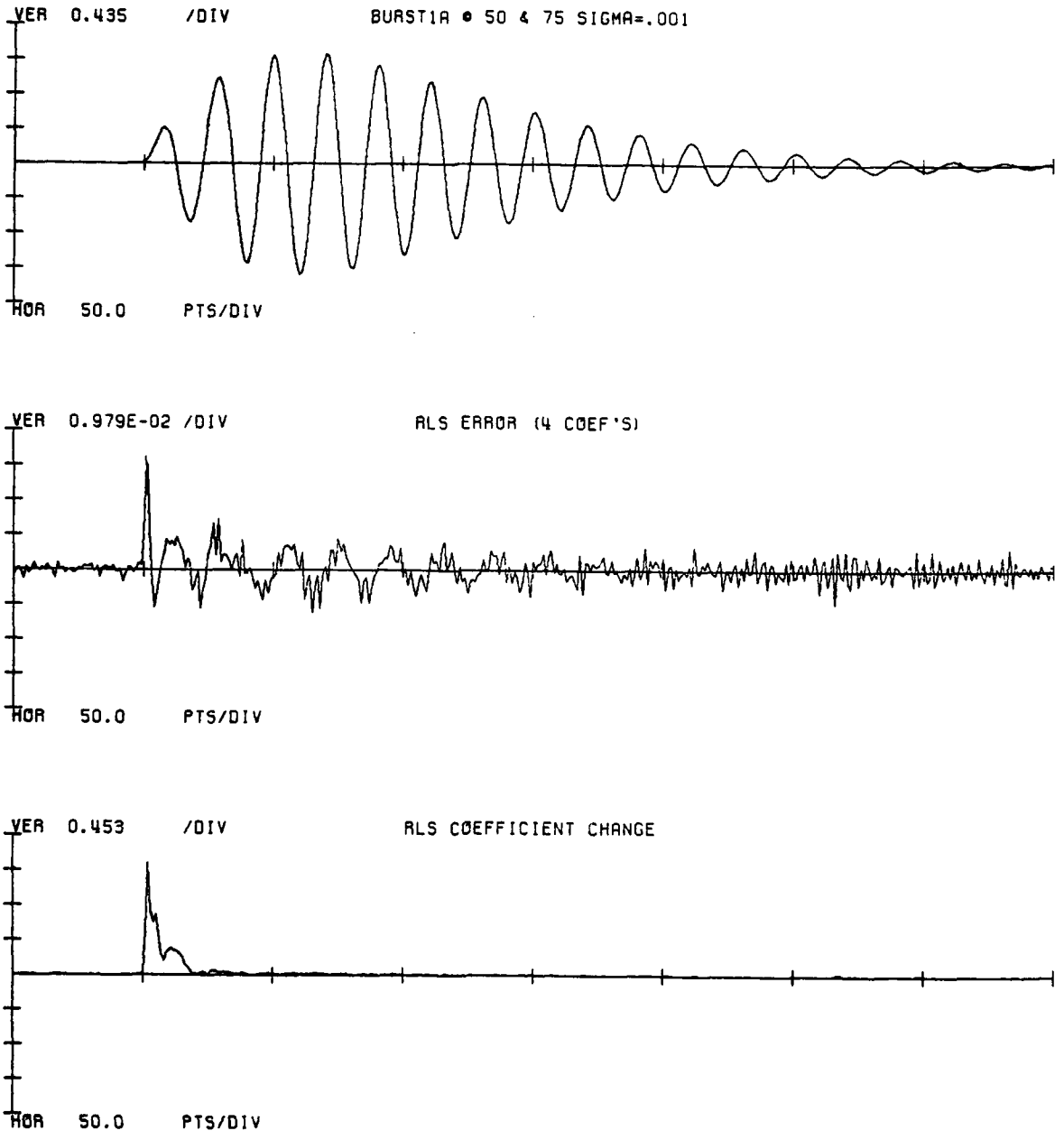

VER 0.435 /DIV
BURST1A @ 50 & 75 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.979E-02 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.453 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.20a

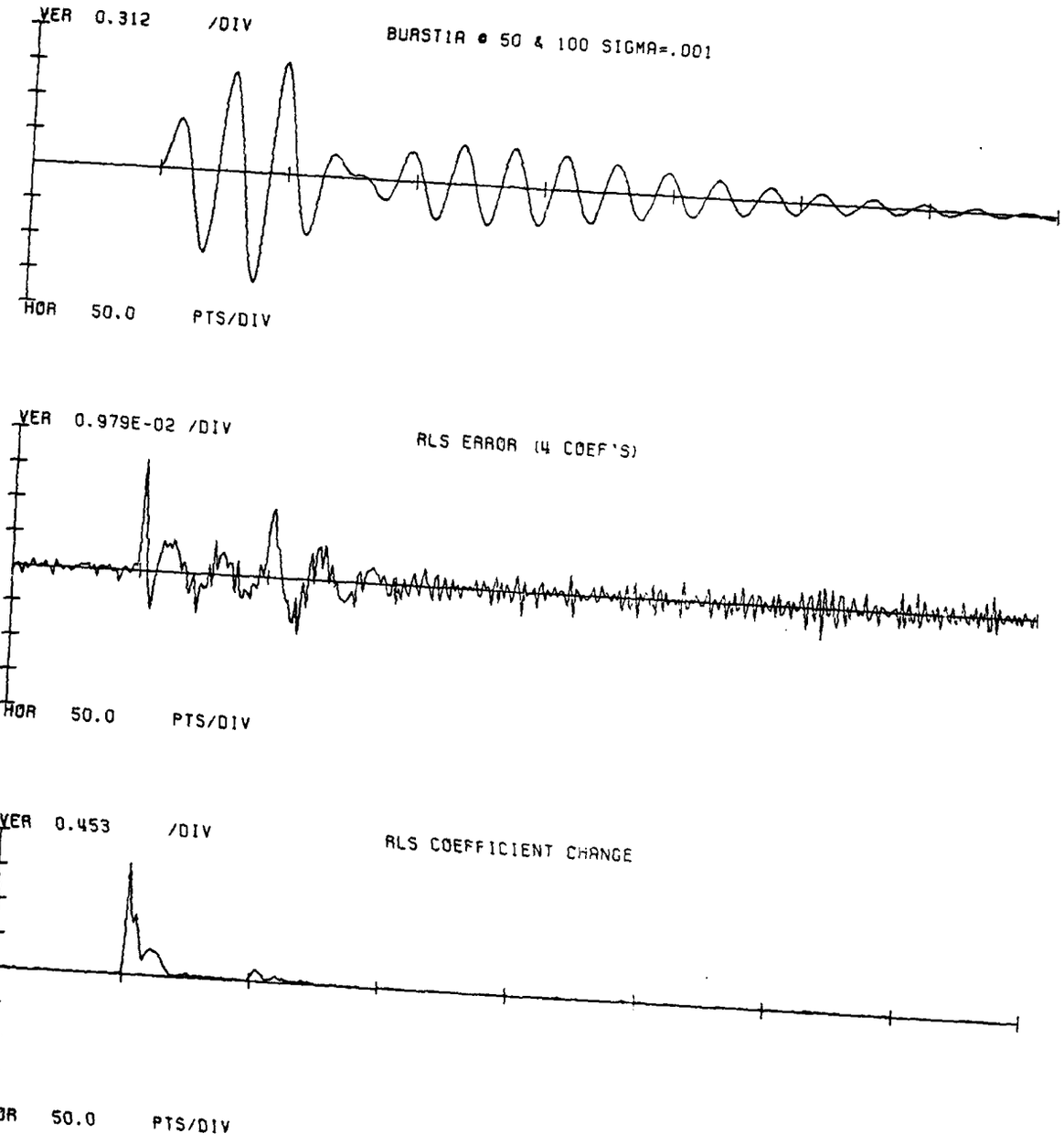
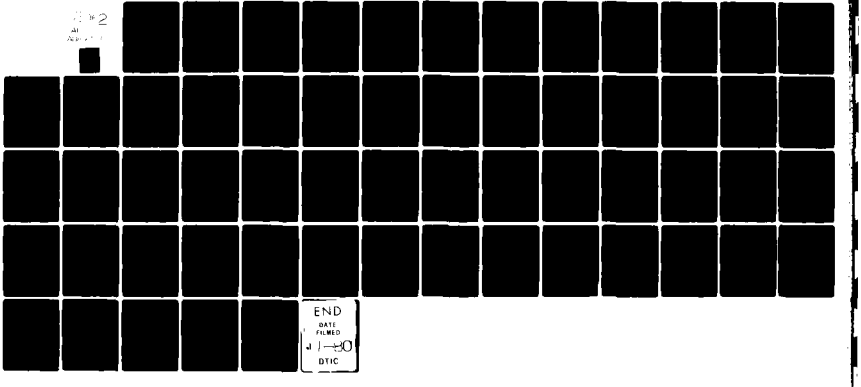
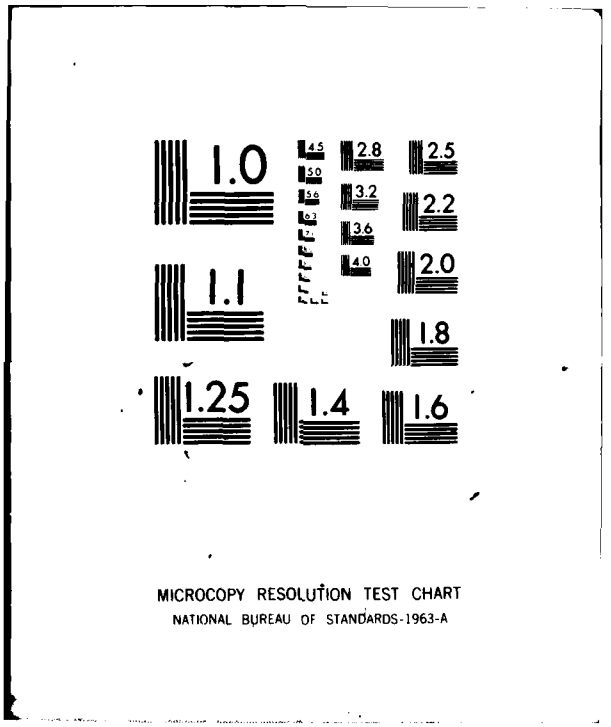

VER 0.312 /DIV
BURSTIR @ 50 & 100 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.979E-02 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.453 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.20b

AD-A089 785 MASSACHUSETTS INST OF TECH CAMBRIDGE RESEARCH LAB OF--ETC F/G 17/9
EVENT COMPRESSION USING RECURSIVE LEAST SQUARES SIGNAL PROCESSI--ETC(U
JUL 80 W P DOVE N00014-75-C-0951
UNCLASSIFIED TR-492 NL





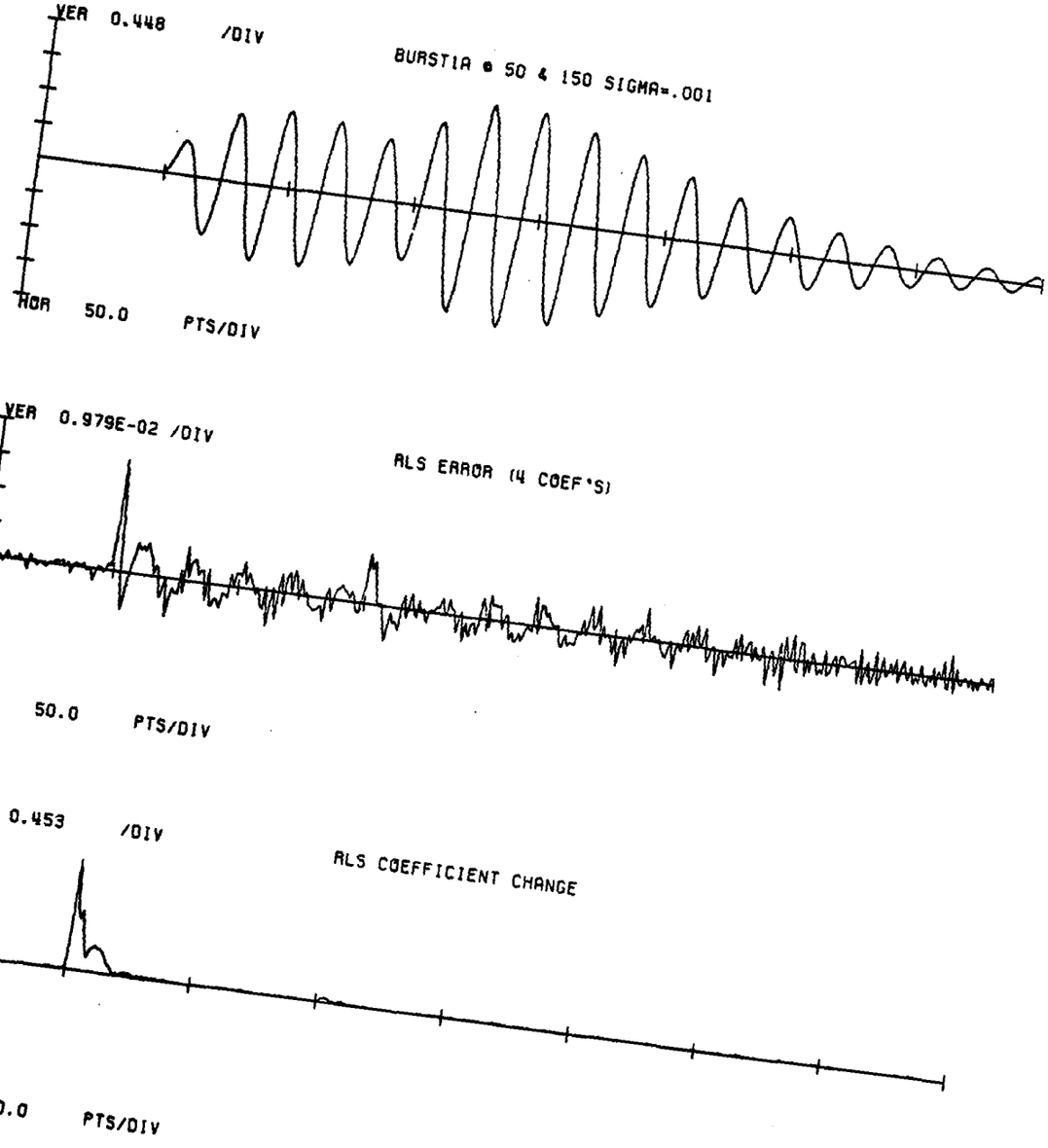

VER 0.448 /DIV
BURSTIA @ 50 & 150 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.979E-02 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.453 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.20c

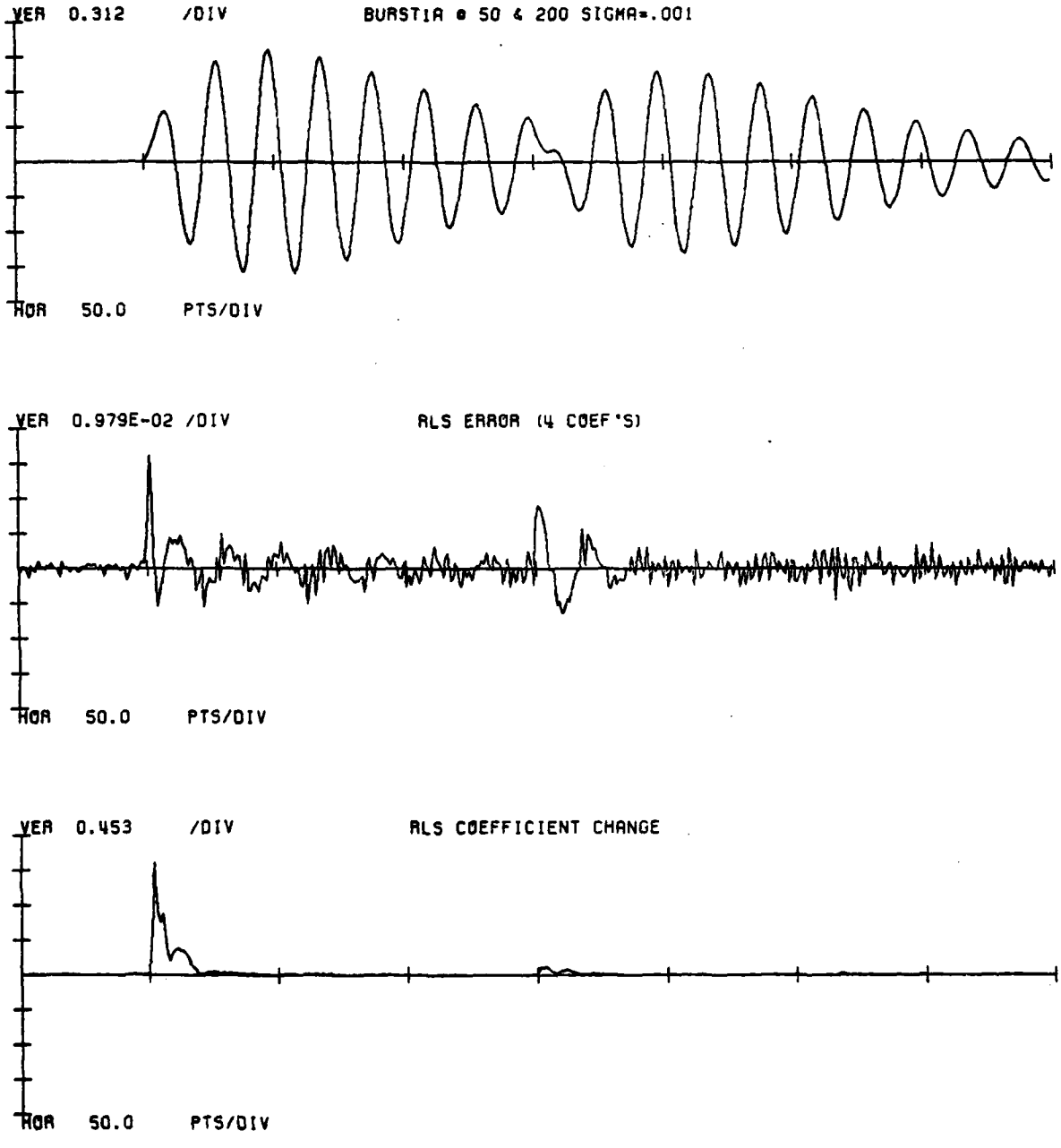
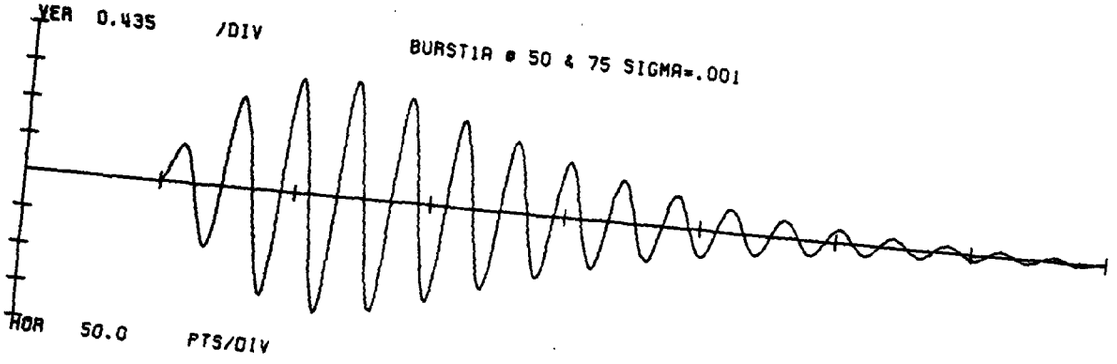
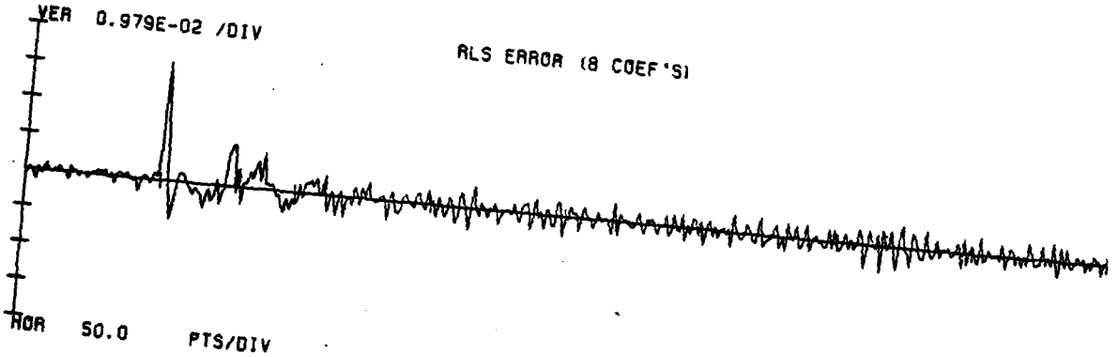

VER 0.312 /DIV
BURSTIA @ 50 & 200 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.979E-02 /DIV
RLS ERROR (4 COEF'S)
HOR 50.0 PTS/DIV
VER 0.453 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.20d


VER 0.435 /DIV
BURST1A @ 50 & 75 SIGMA=.001
HOR 50.0 PTS/DIV


VER 0.979E-02 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV

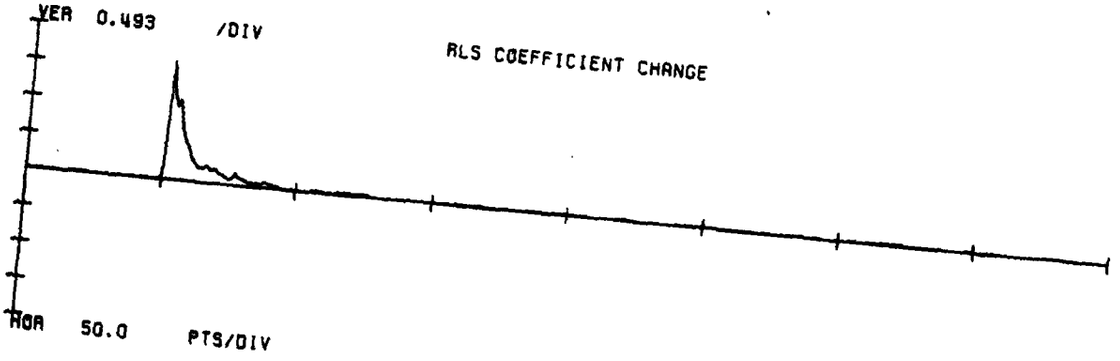

VER 0.493 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.21a

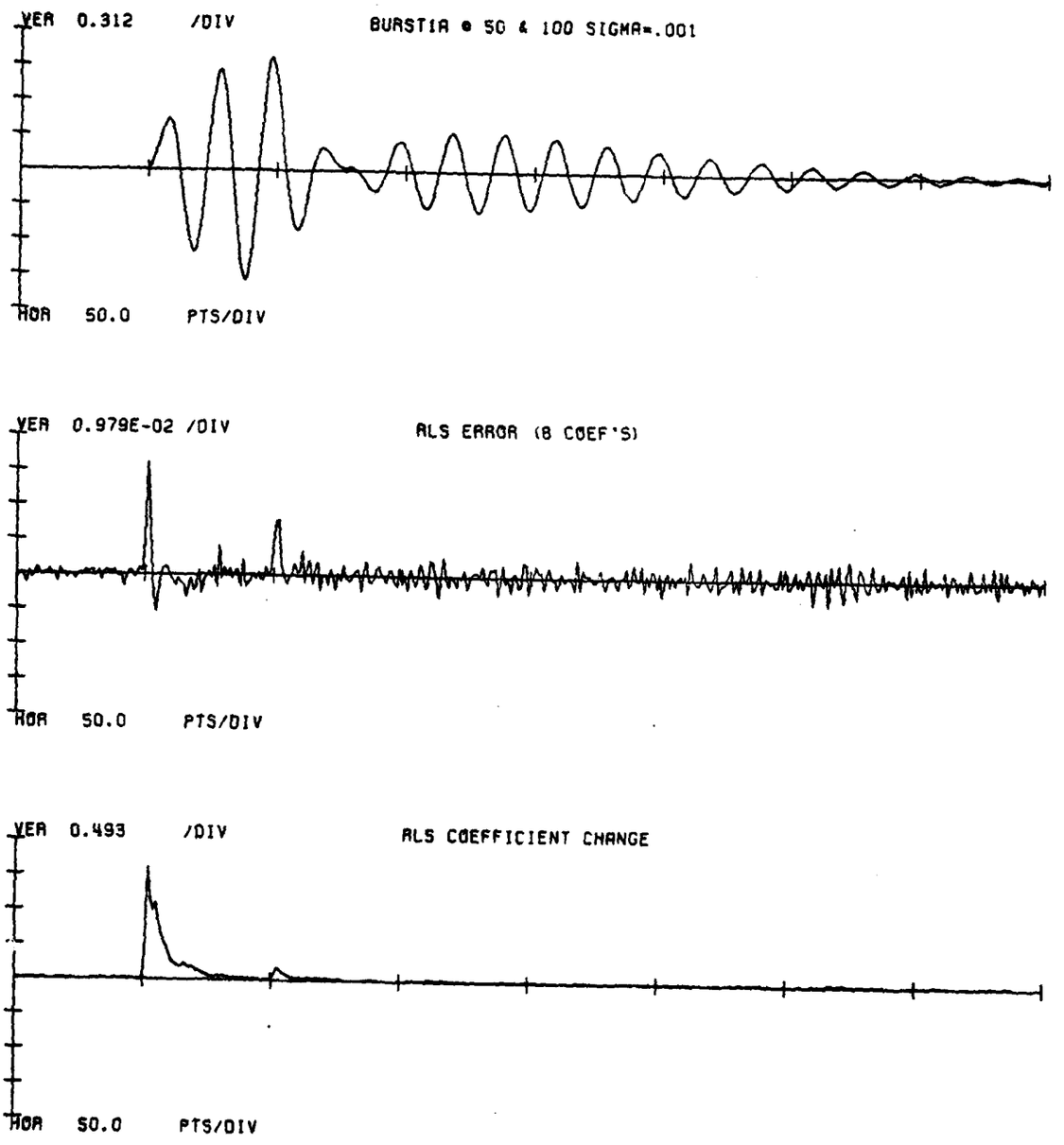
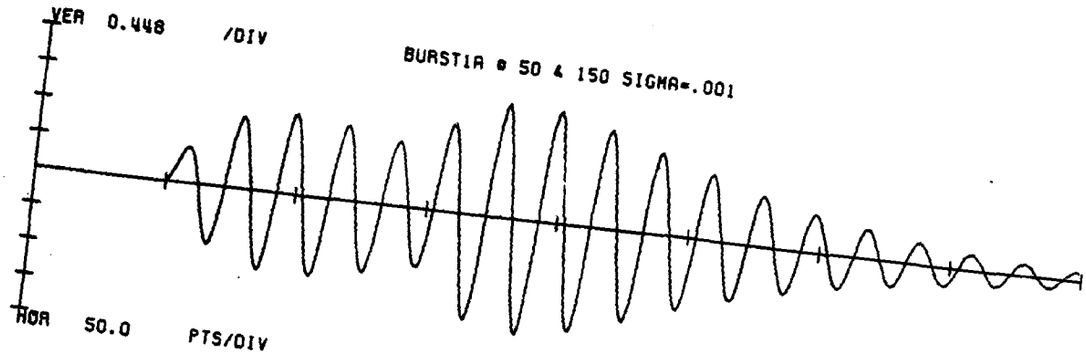
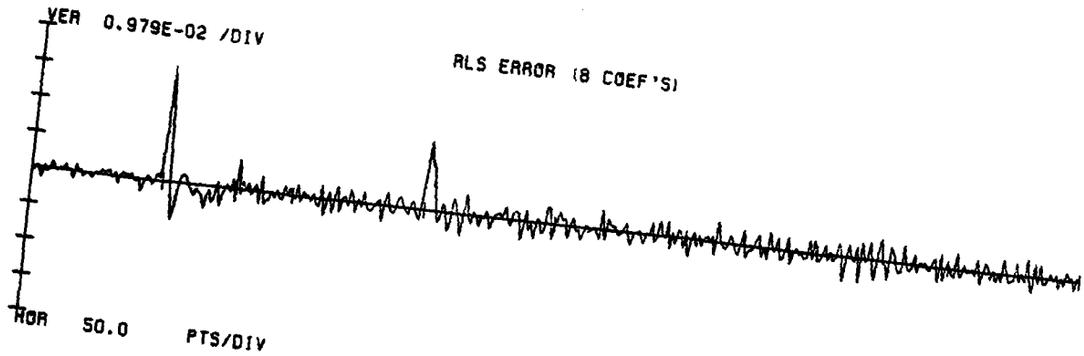

VER 0.312 /DIV
BURSTIR @ 50 & 100 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.979E-02 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV
VER 0.493 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.21b


VER 0.448 /DIV
BURSTIA @ 50 & 150 SIGMA=.001
HOR 50.0 PTS/DIV


VER 0.979E-02 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV

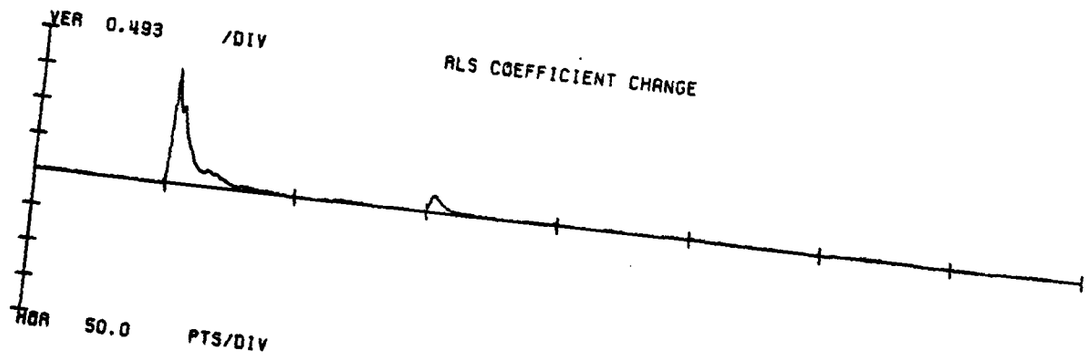
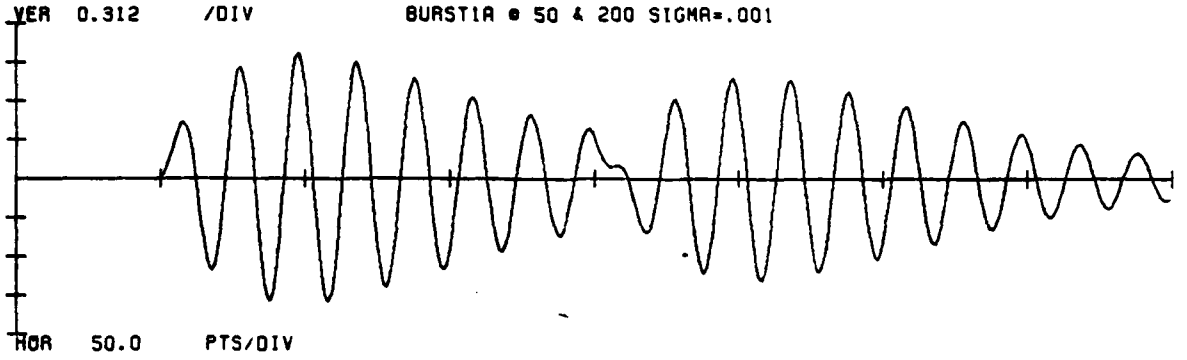
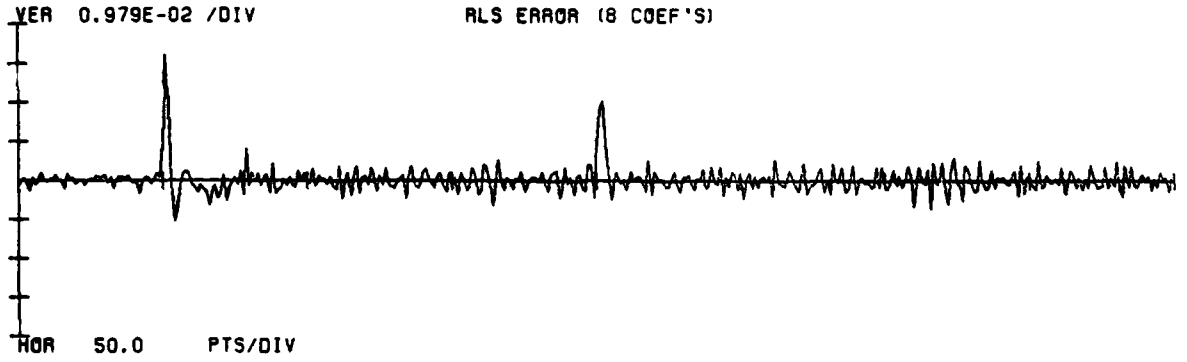

VER 0.493 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.21c


VER 0.312 /DIV
BURST1A @ 50 & 200 SIGMA=.001
HOR 50.0 PTS/DIV


VER 0.979E-02 /DIV
RLS ERROR (8 COEF'S)
HOR 50.0 PTS/DIV

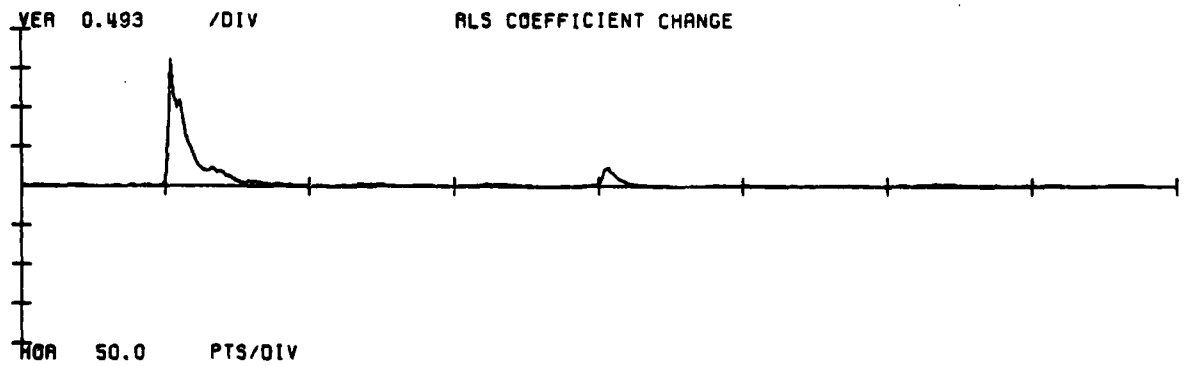

VER 0.493 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.21d

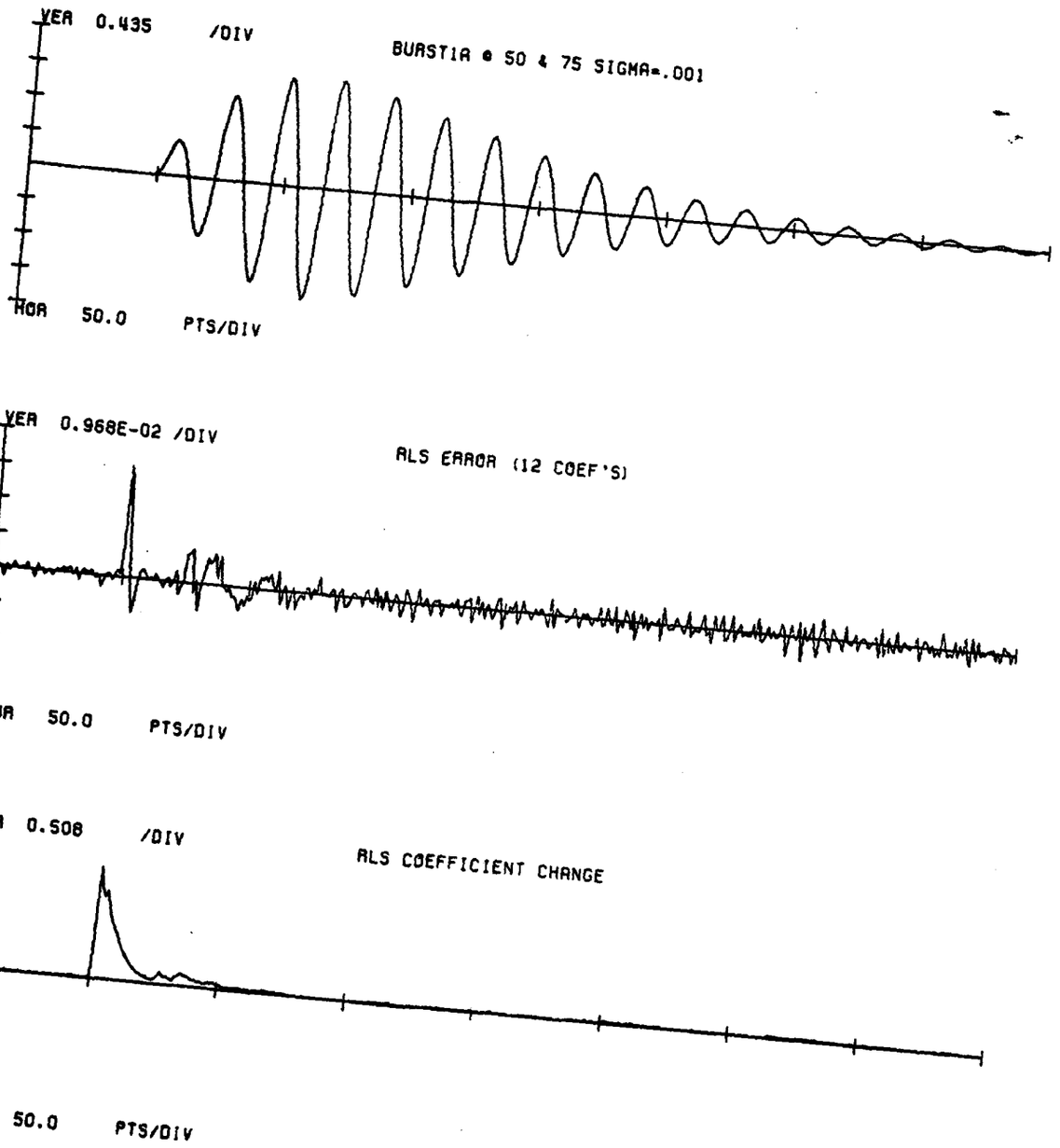

VER 0.435 /DIV
BURSTIR @ 50 & 75 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.968E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.508 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.22a

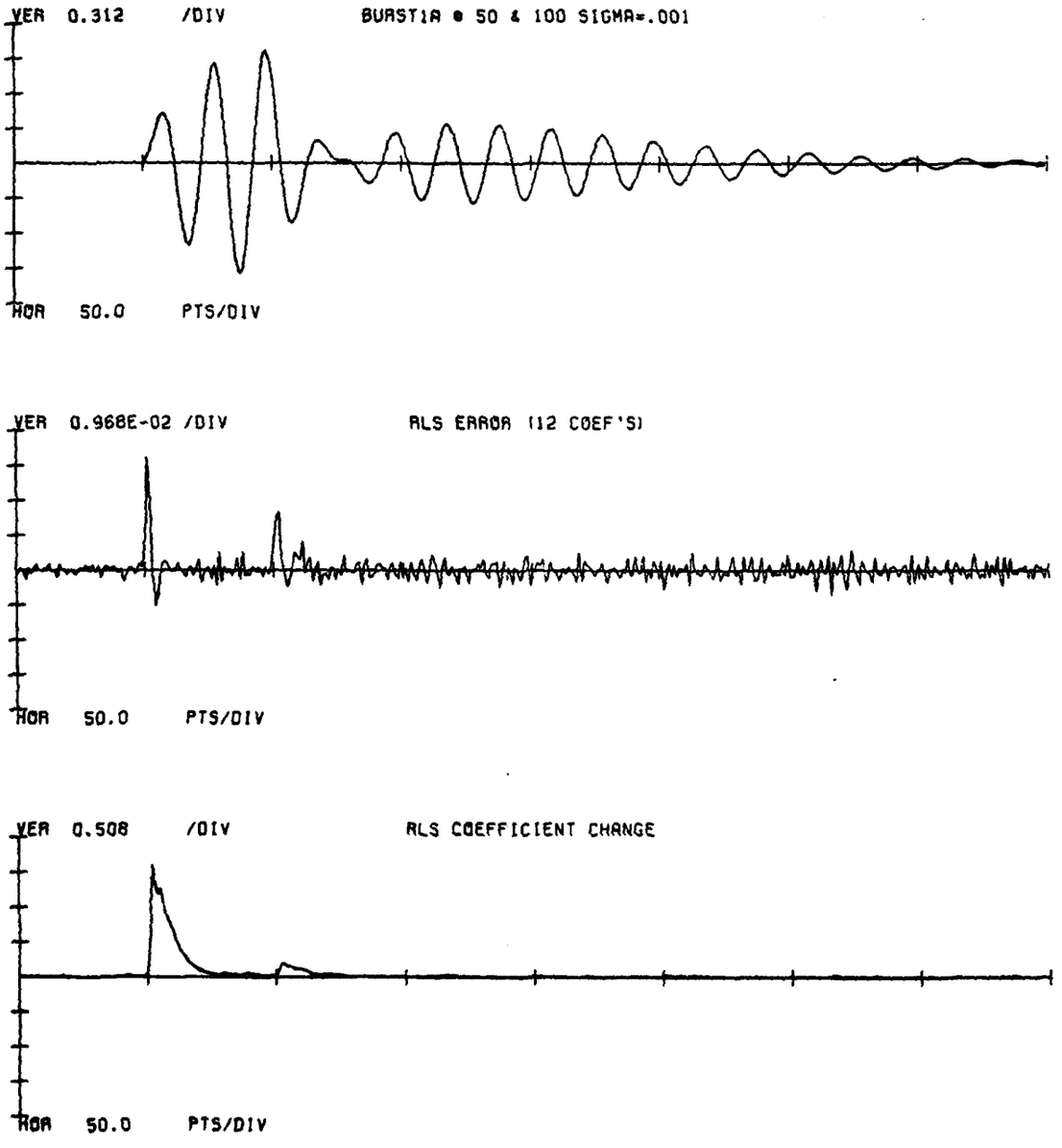

VER 0.312 /DIV
BURST1A @ 50 & 100 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.968E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.508 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.22b

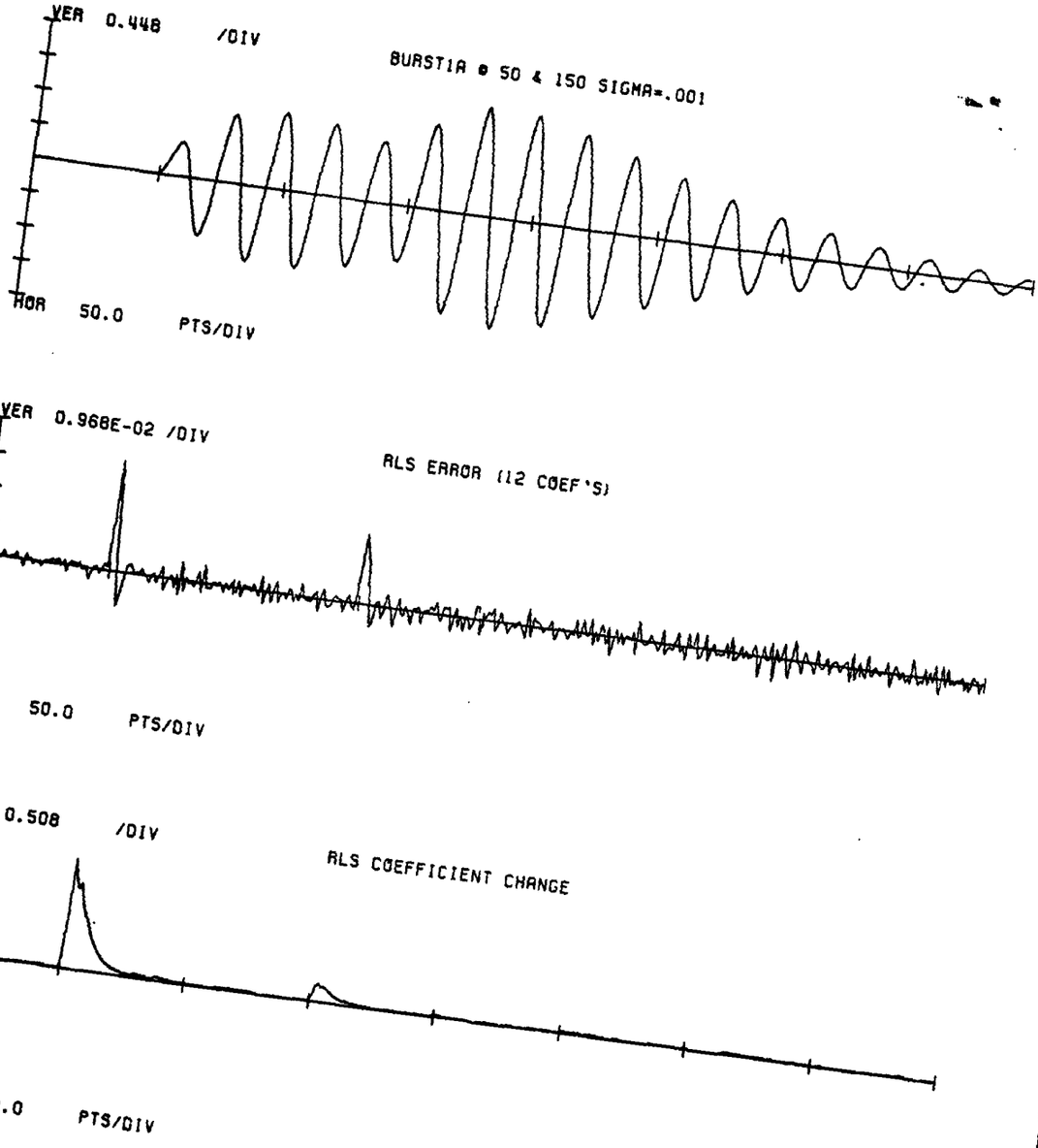
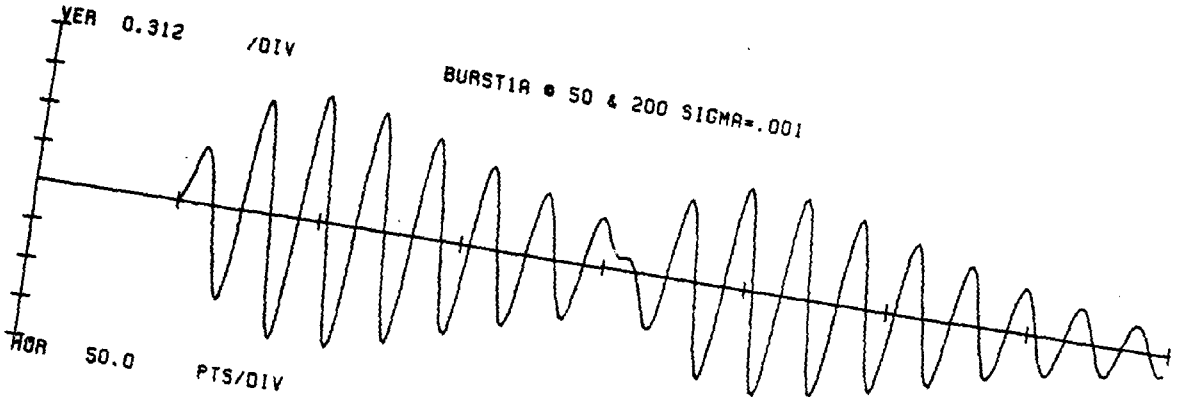
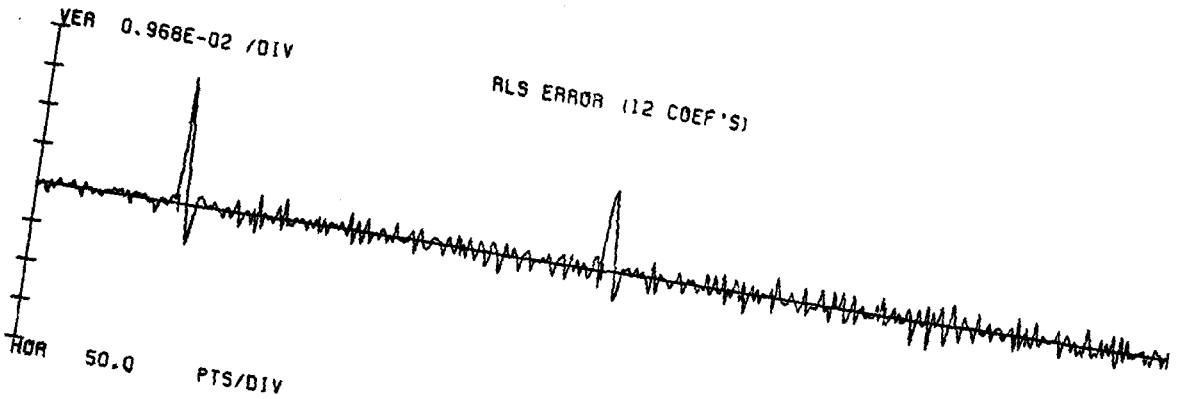

VER 0.448 /DIV
BURST1A @ 50 & 150 SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.968E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.508 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.22c


VER 0.312 /DIV
BURST1A @ 50 & 200 SIGMA=.001
HOR 50.0 PTS/DIV


VER 0.968E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV

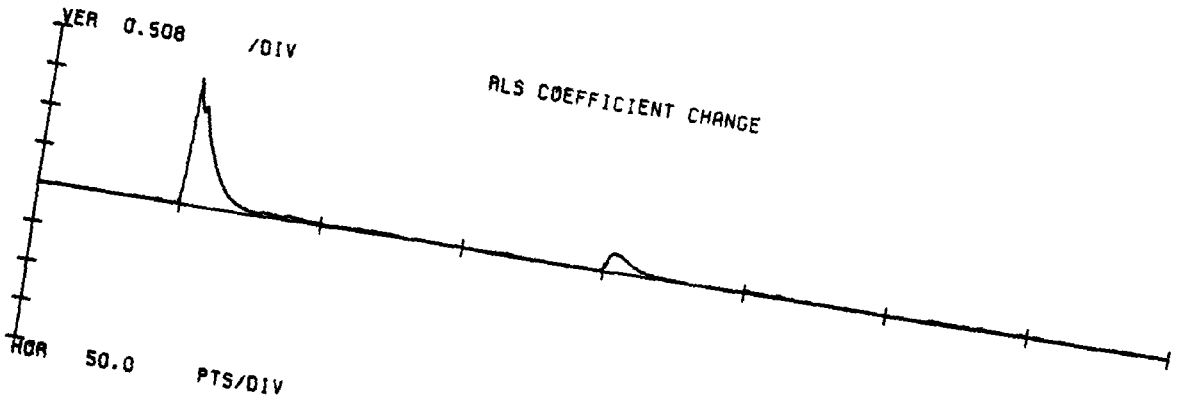

VER 0.508 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.22d

BURSTIA @ 100

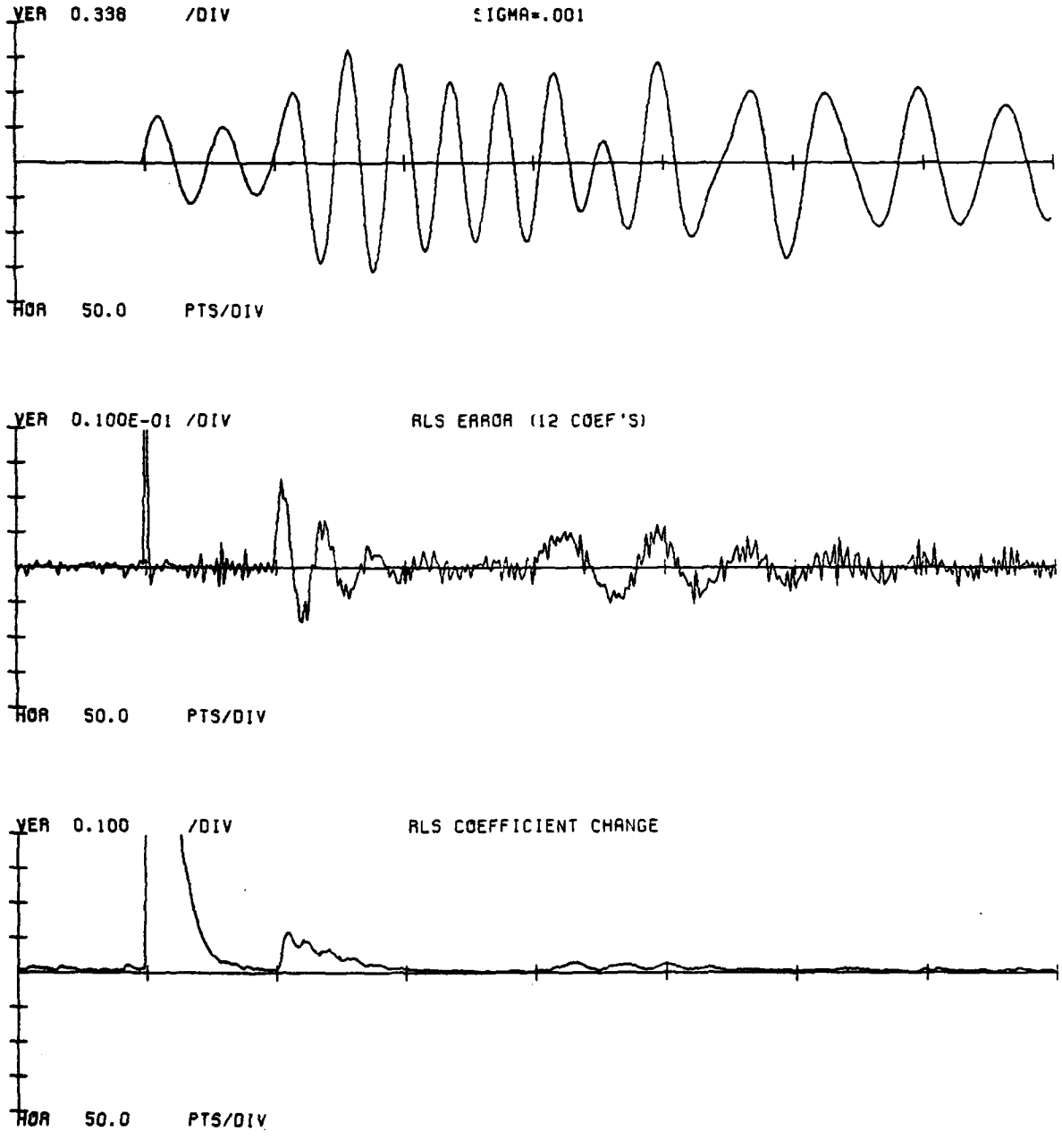


Figure 3.23a

BURST1A @ 150

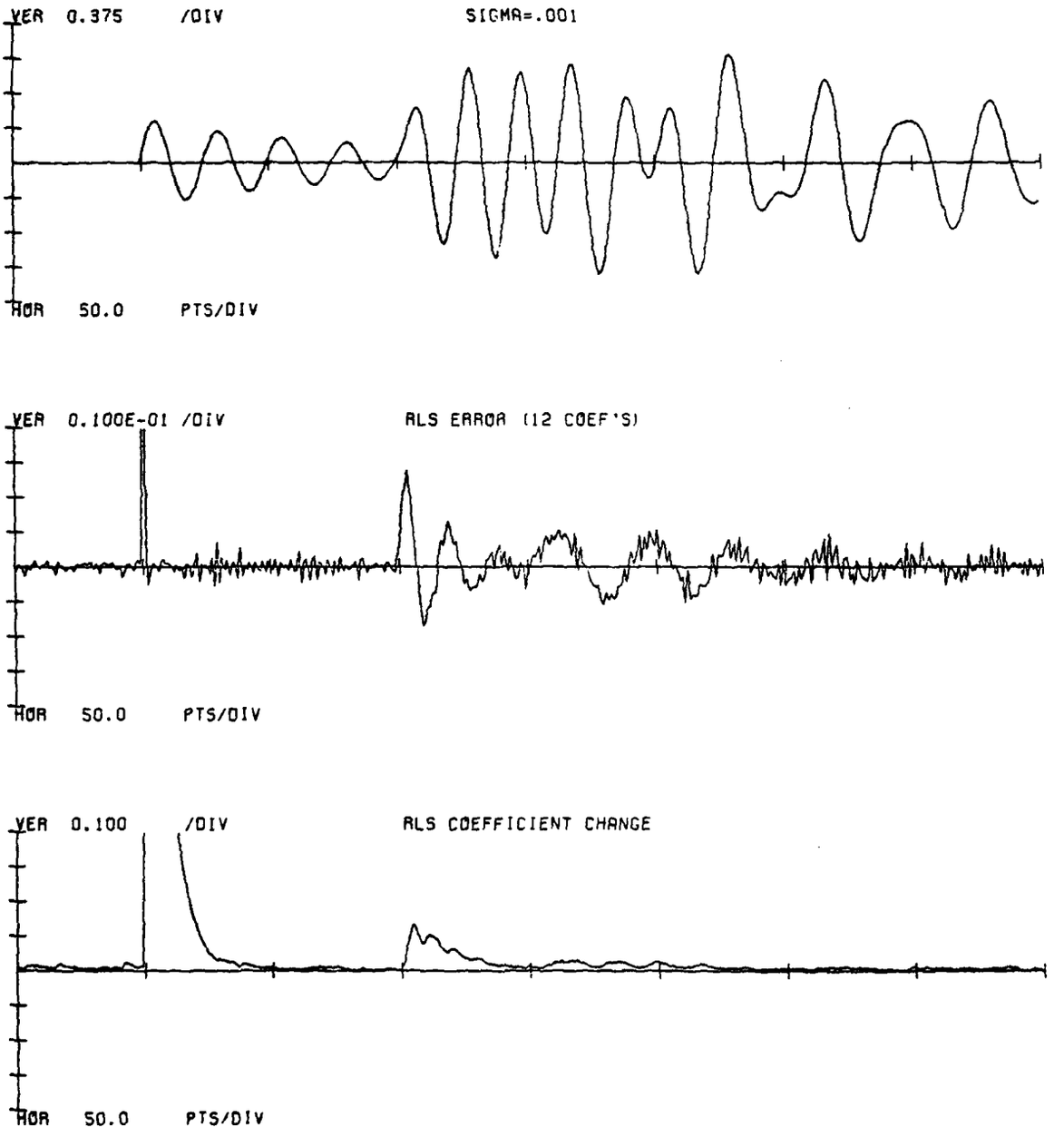


Figure 3.23b

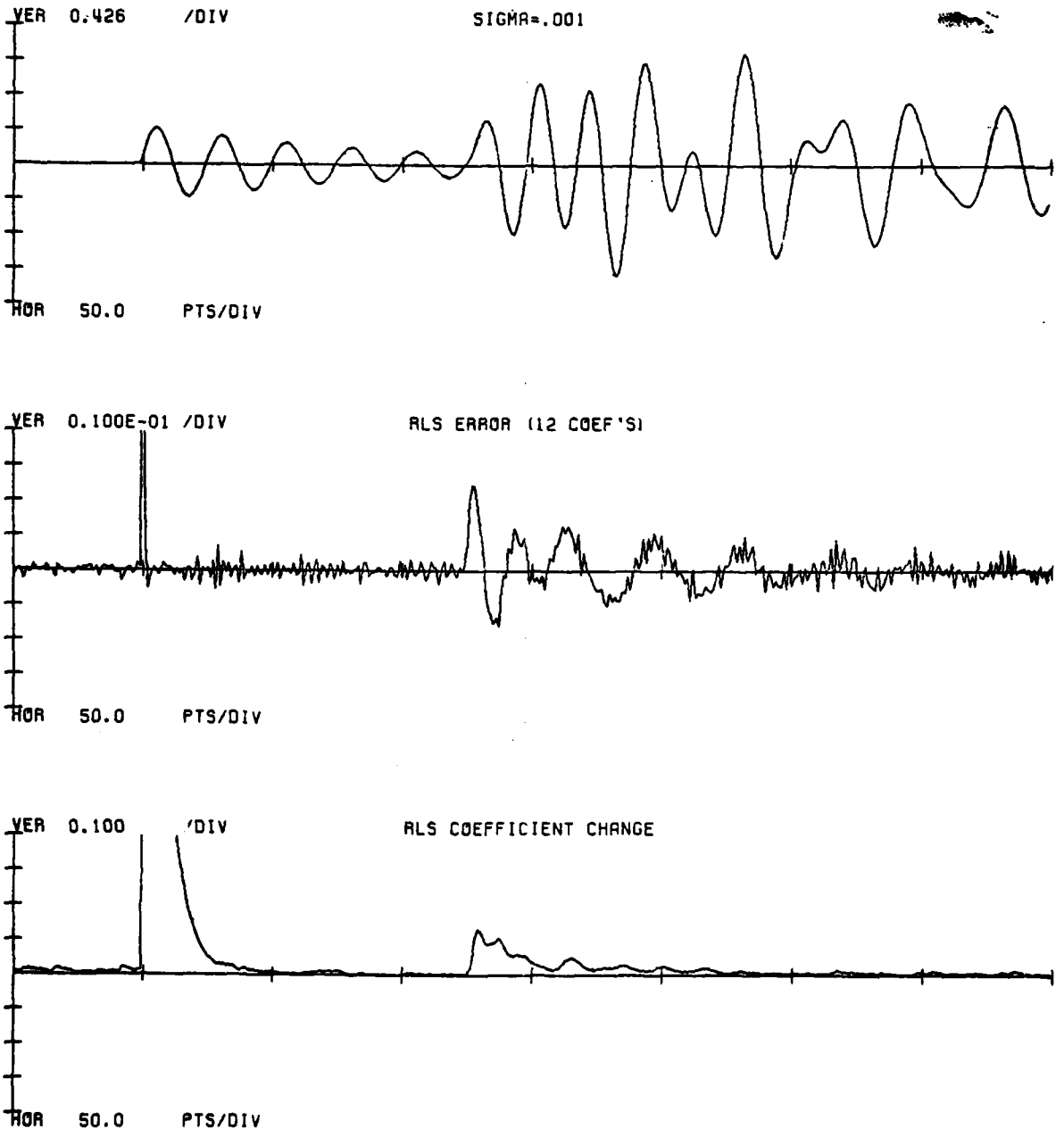

VER 0.426 /DIV
SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.100E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.100 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.23c

BURST1A @ 200

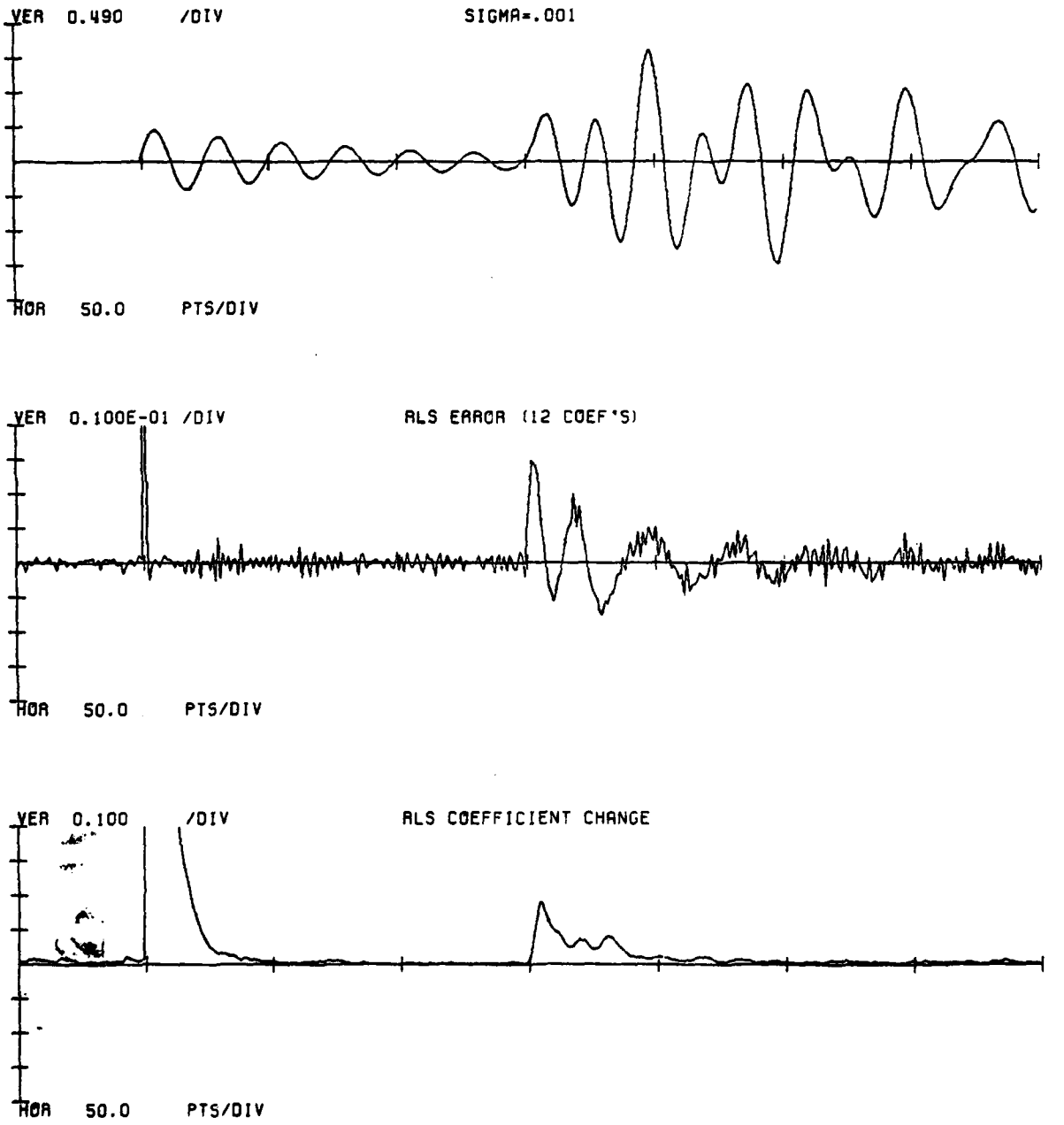


Figure 3.23d

BURST1A @ 225

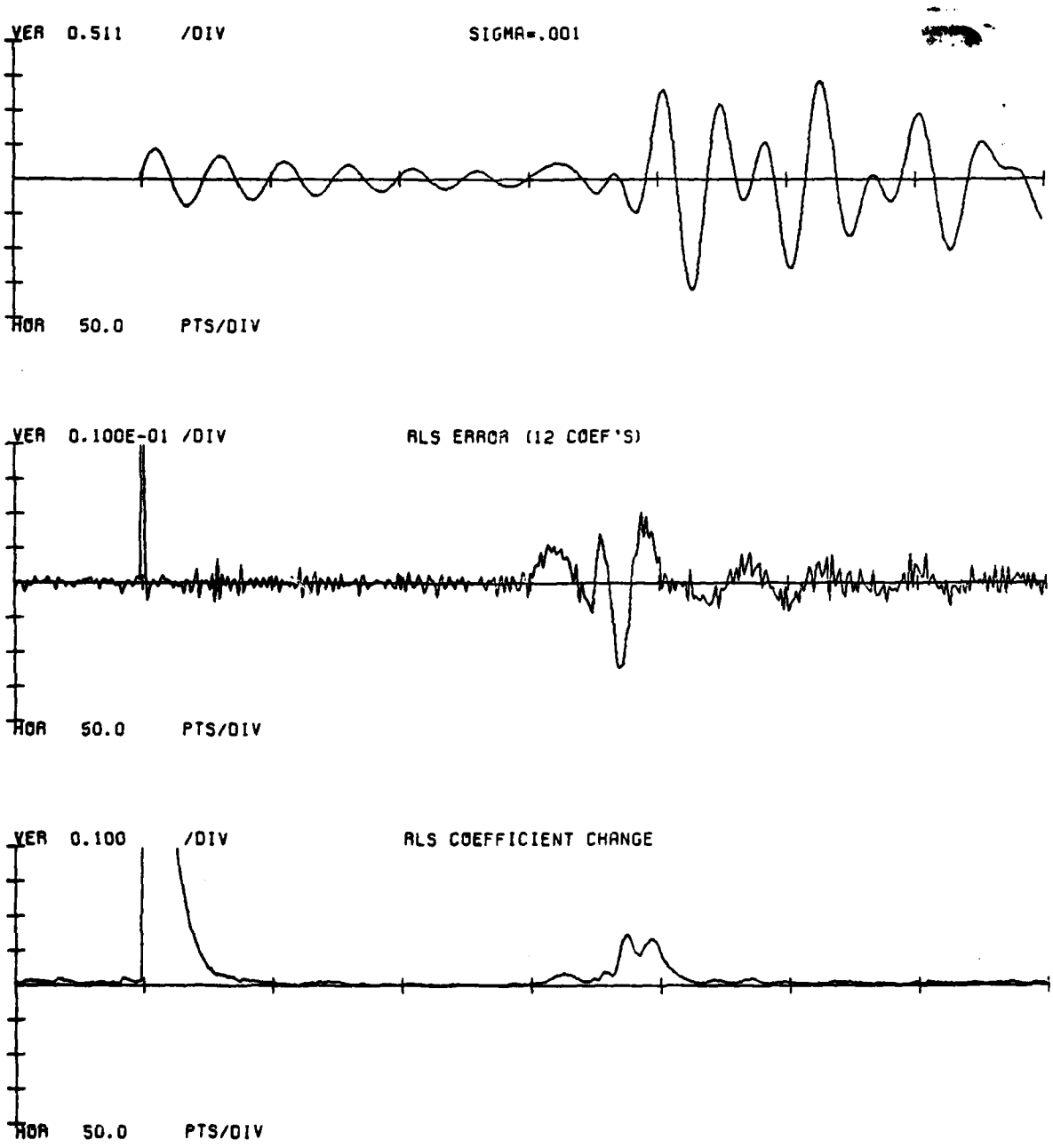


Figure 3.23e

BURST1A @ 250

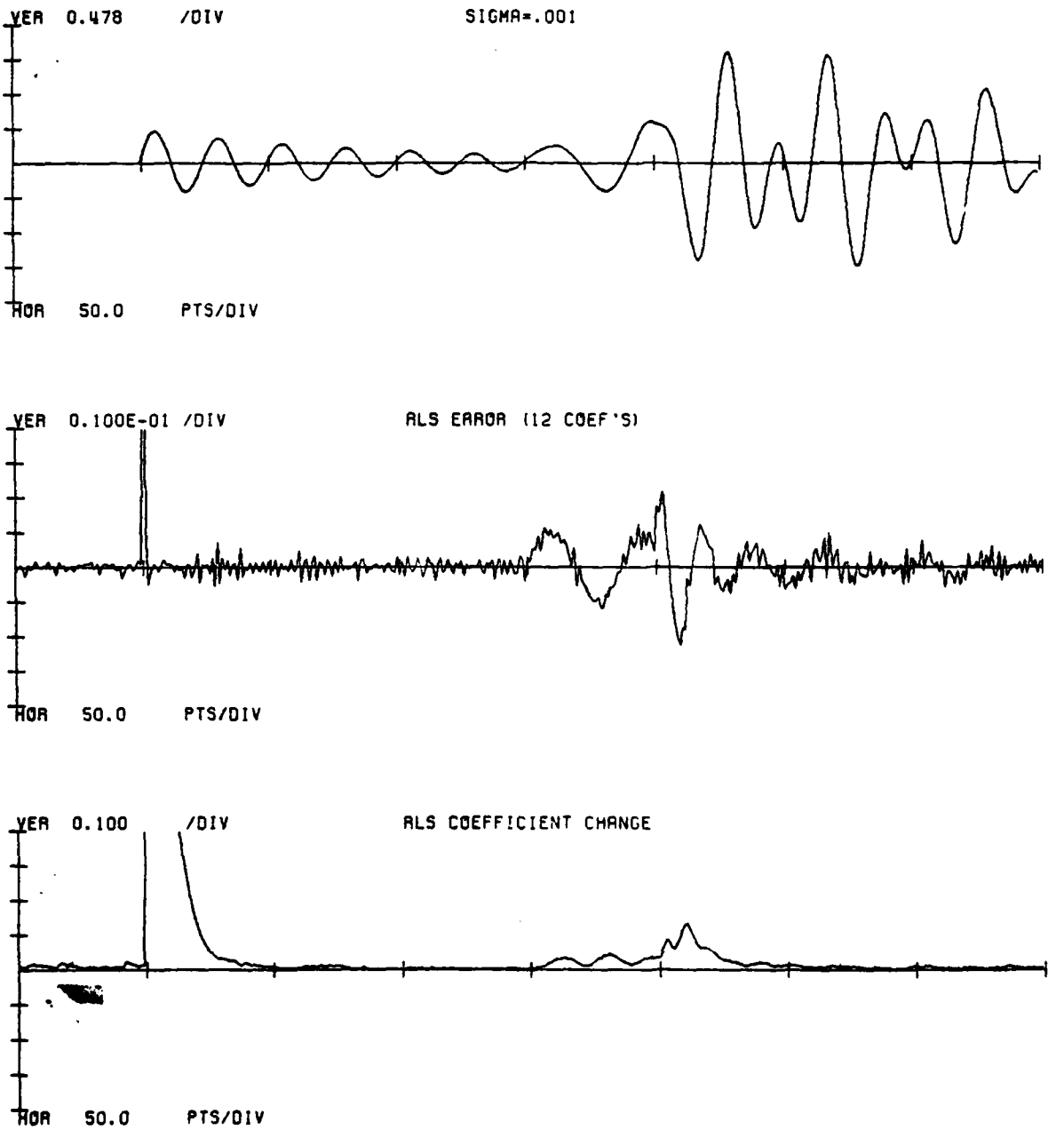


Figure 3.23f

BURST1A @ 300

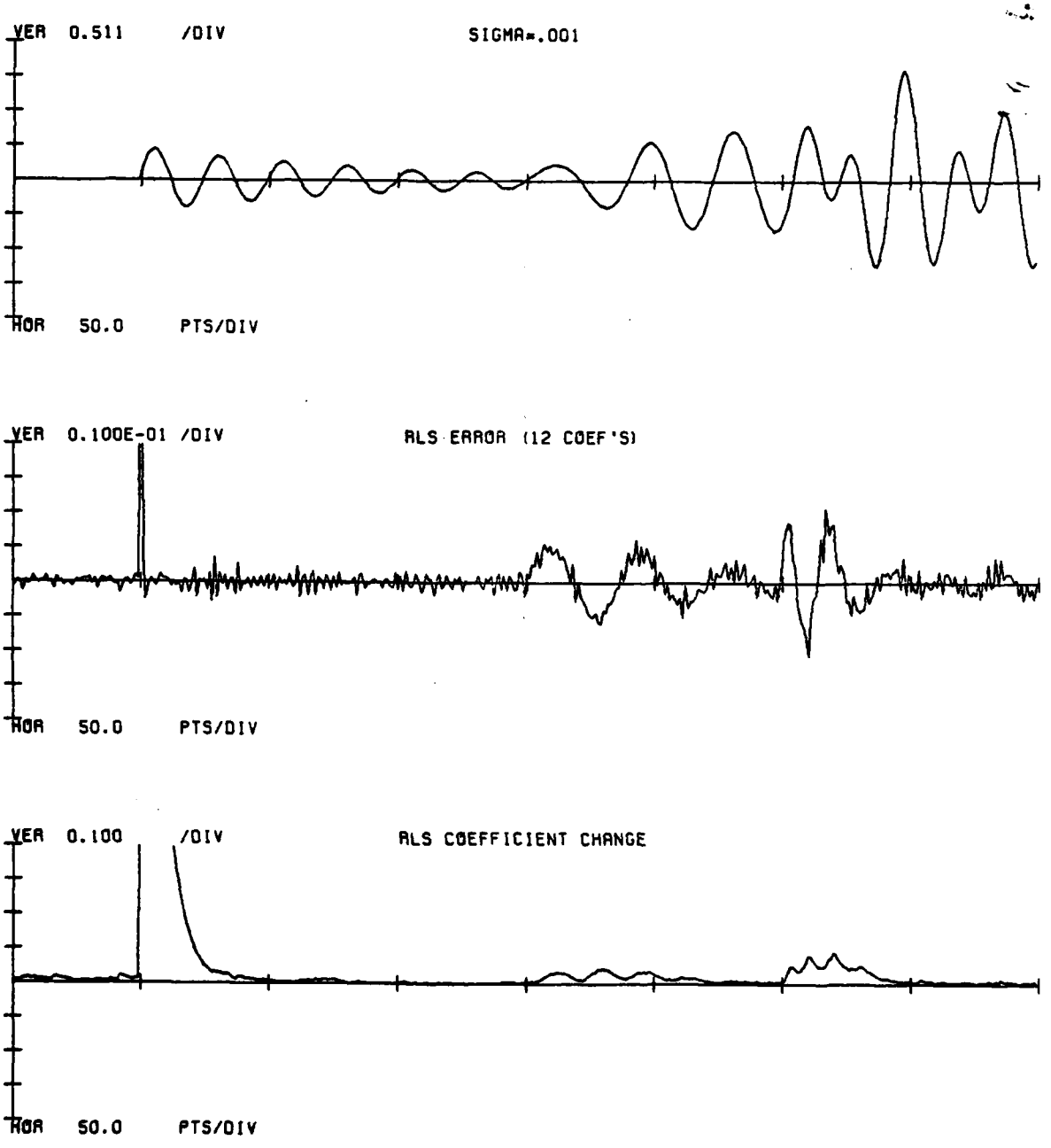


figure 3.23g

BURST1A ● 350

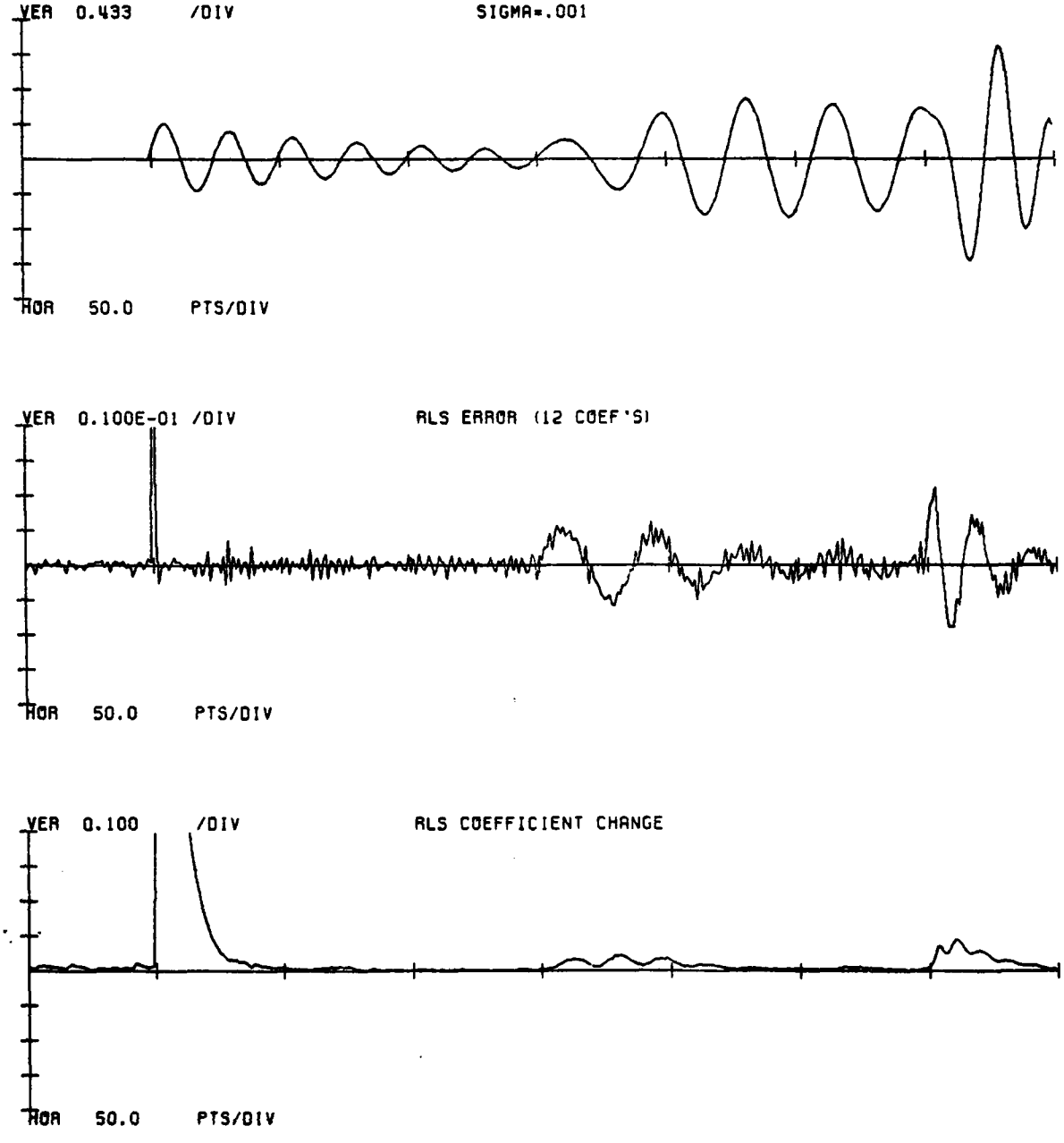


Figure 3.23h

3.4 Linear Filtering

It should be evident from the preceding sections that noise degrades these event compression signals. To try enhancing the quality of these signals (on noisy data) experiments were performed using linear filtering as a pre- and post-process to event compression with RLS. Three questions were examined:

> Does prefiltering the data to reduce noise improve the quality of the observed compressed events?
>
> Is all-pole filtering preferable to FIR filtering for this application?
>
> Will postfiltering the RLS error make events in it more visible?

We decided to prefilter the data because we thought that reducing the high frequency noise in the data might allow the predictor to adapt more quickly to the events and thereby produce sharper events in the coefficient change signal and faster settling in the RLS error. The spectra of the Burst1 signal with three different noise levels are shown in figures 3.24a-3.24c. It is evident from these figures that the frequency band from $.1f_{sample}$ to $.5f_{sample}$ is dominated by noise. We thought that reducing the noise in this band would reduce the error energy cost $\mathbf{d}^T\mathbf{R}\mathbf{d}$ of changing the predictor (see eq. 3.5) and thereby permit more responsiveness to the events in the data.

Two filters were designed for the purpose of reducing the noise power in the frequency band extending from $.1f_{sample}$ to $.5f_{sample}$. Figure 3.25 illustrates the time response and spectrum of a 50 point FIR lowpass filter designed with the Parks-McClellan algorithm [McClellan et. al.,1973]. Figure 3.26 shows the denominator coefficients and inverse spectrum of a 6 point purely recursive (all-pole) filter generated by the minimum p criterion IIR filter design program [Deczky,1972]. Figures 3.27a-c demonstrate the effect of FIR prefiltering on the event compressed signals.[1] Figures 3.28a-c show the effects of IIR (purely recursive) filtering.

The first FIR filtered example contains extended events in the compressed signals due to inadequate prediction. FIR filtering convolves the input sequence with the sequence given in figure 3.25 . The convolution adds 49 zeroes to each of the bursts making them difficult to model via an all-pole technique such as RLS. This causes the 50 point long bursts in the RLS error and coefficient change for the low noise examples. (Surprisingly, the event corresponding to the second burst seems unaffected.) The fact that this effect is not apparent in the noisy examples is currently not

---

1. These figures have been adjusted to position the events in the same places as those in the IIR example. That is, they have been left shifted 25 points.

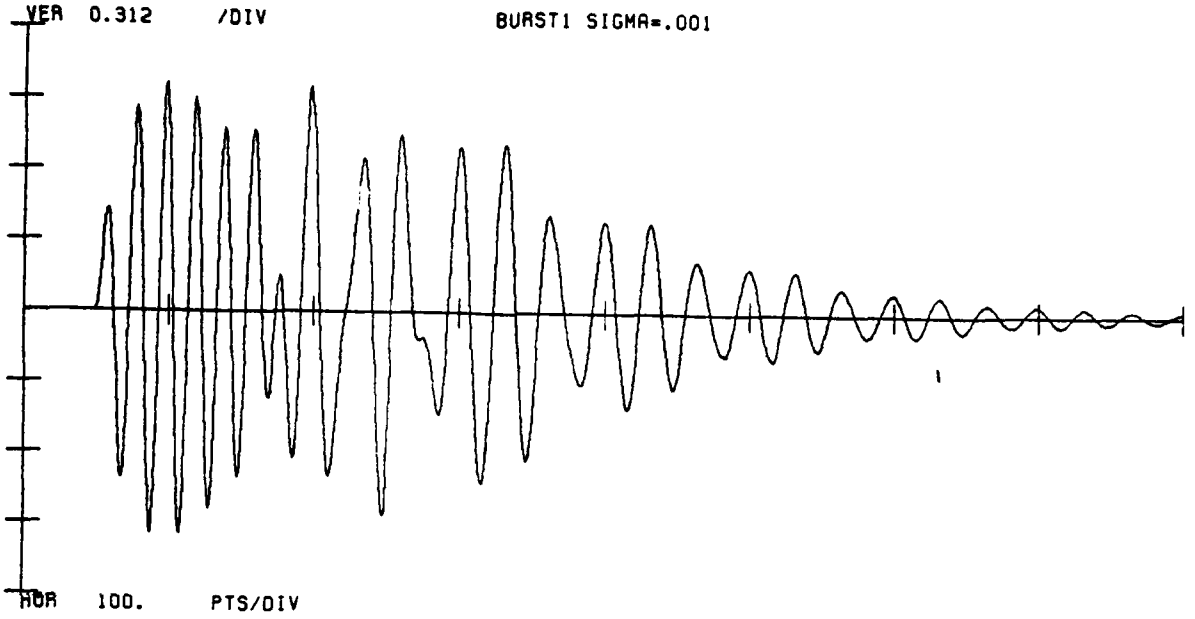
understood.

The IIR filtering process does not introduce a large number of zeroes to each burst, instead it adds 5 poles. Thus, the resulting data is more easily modelled by the RLS algorithm than in the FIR case (since it is an all-pole modelling technique) and therefore the event location signals are better behaved.

Unfortunately, comparison of these figures with figure 3.17 indicates that prefiltering by either FIR or all-pole filters does not improve the quality of the RLS error signal events; and both types of filtering cause an undesirable increase in the noise of the coefficient change signal. The reason for this increase in predictor activity may be that after filtering, the noise can be more effectively predicted because of its increased correlation, and the coefficients change more in trying to predict it. In any case, this method of enhancement seems to be inneffective.

The noise in the RLS error signal obscures the location events therein. To try removing it (and thereby enhance the events) the FIR filter shown in figure 3.25 was applied to the RLS error signal. The results are given in figure 3.29 for various noise levels. In comparison with the unfiltered RLS error signal presented in figure 3.17, there is some reduction of the noise level, but the events themselves

are smeared. Possibly a matched filter could be designed to compress the events in the RLS error, but at this time the characteristics of those events are not known well enough to design such a filter.


VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 100. PTS/DIV

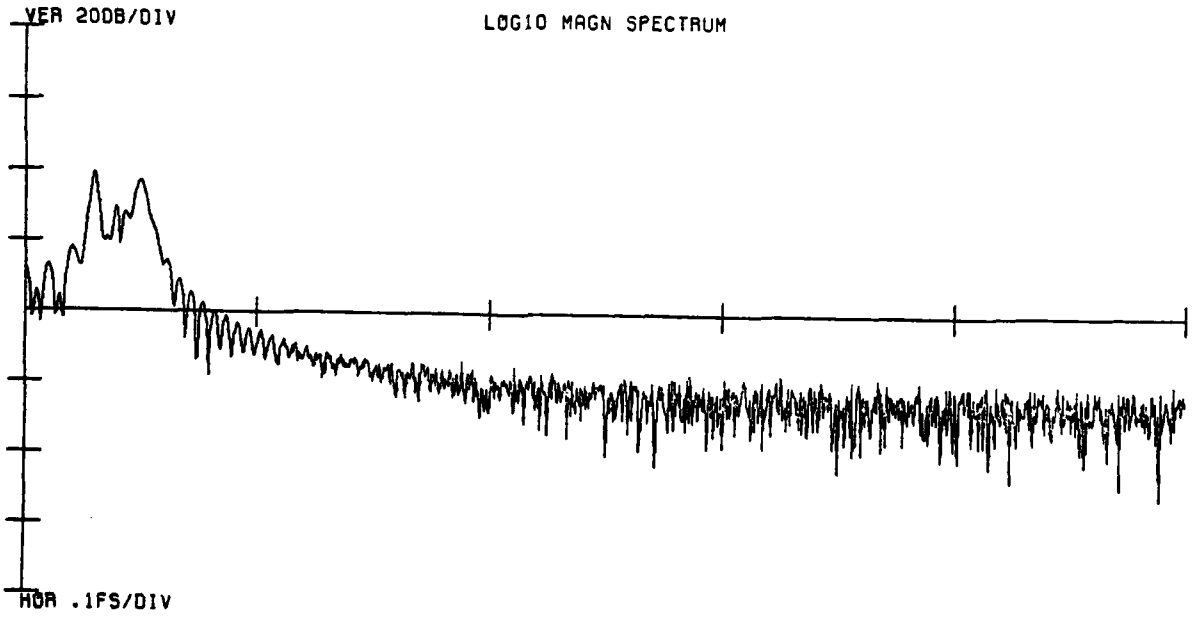
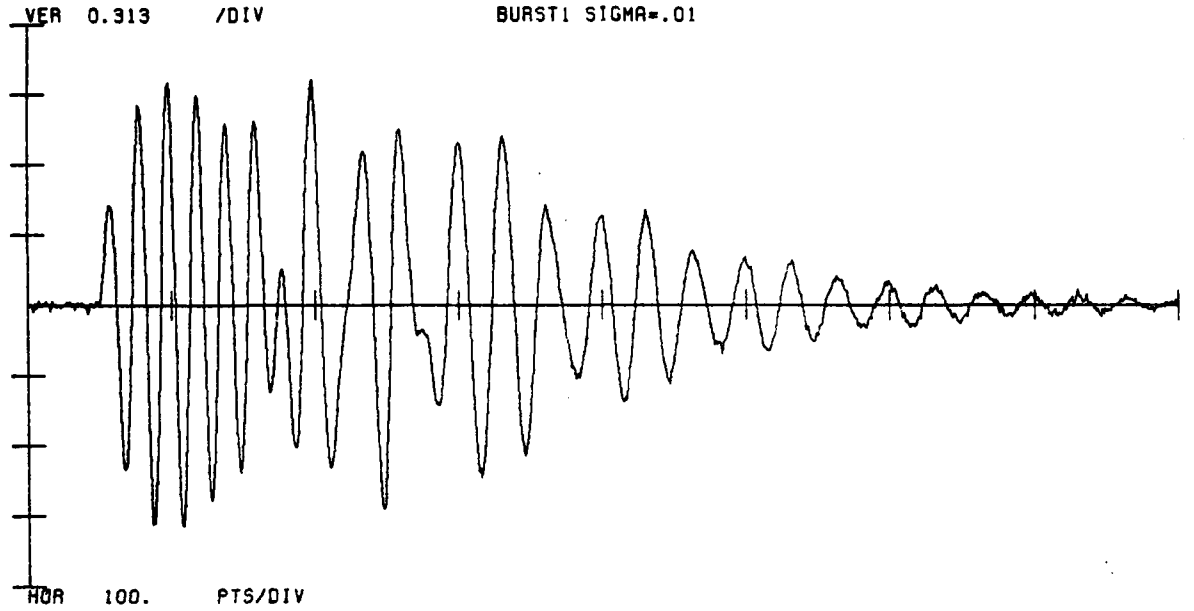

VER 20DB/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.24a


VER 0.313 /DIV
BURST1 SIGMA=.01
HOR 100. PTS/DIV

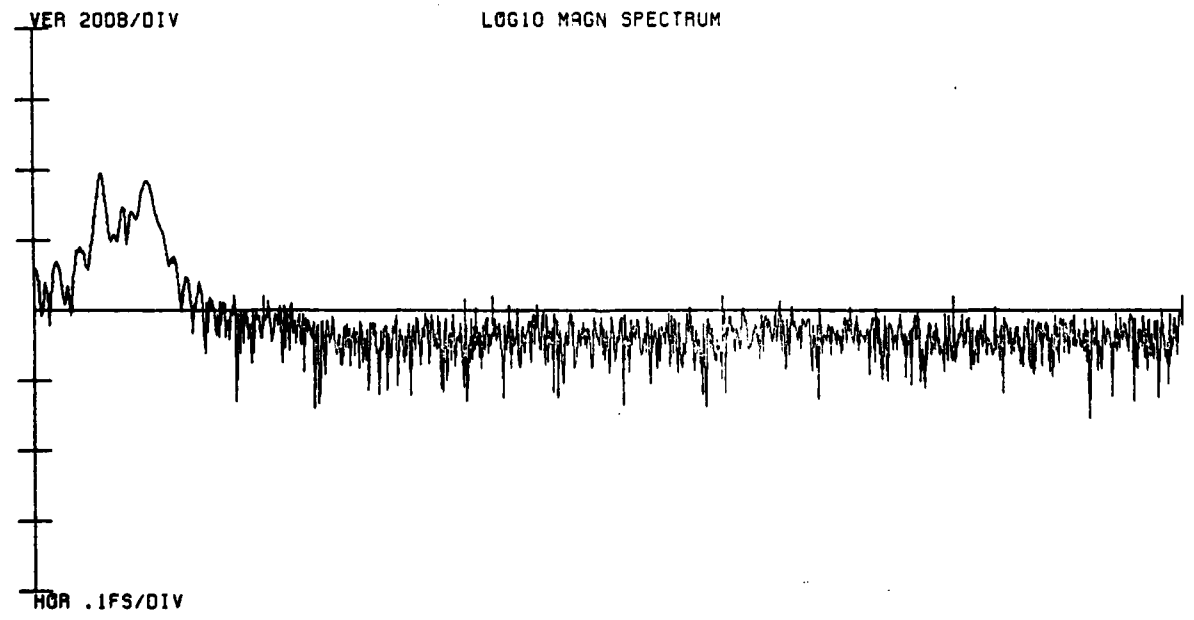
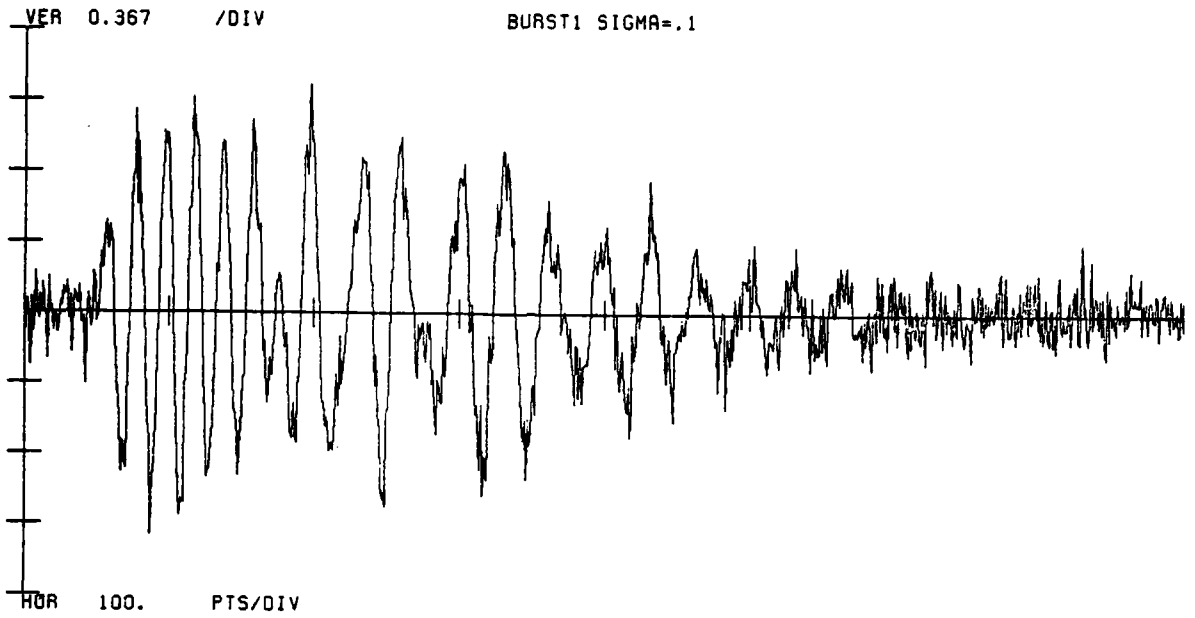

VER 2008/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.24b


VER 0.367 /DIV
BURST1 SIGMA=.1
HOR 100. PTS/DIV

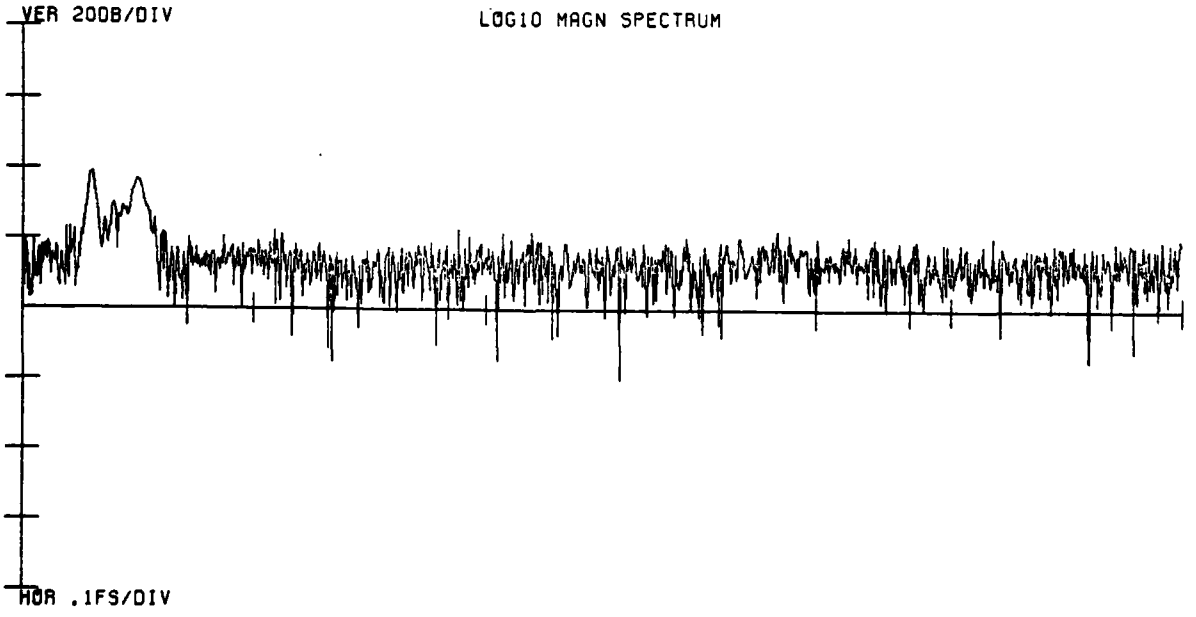
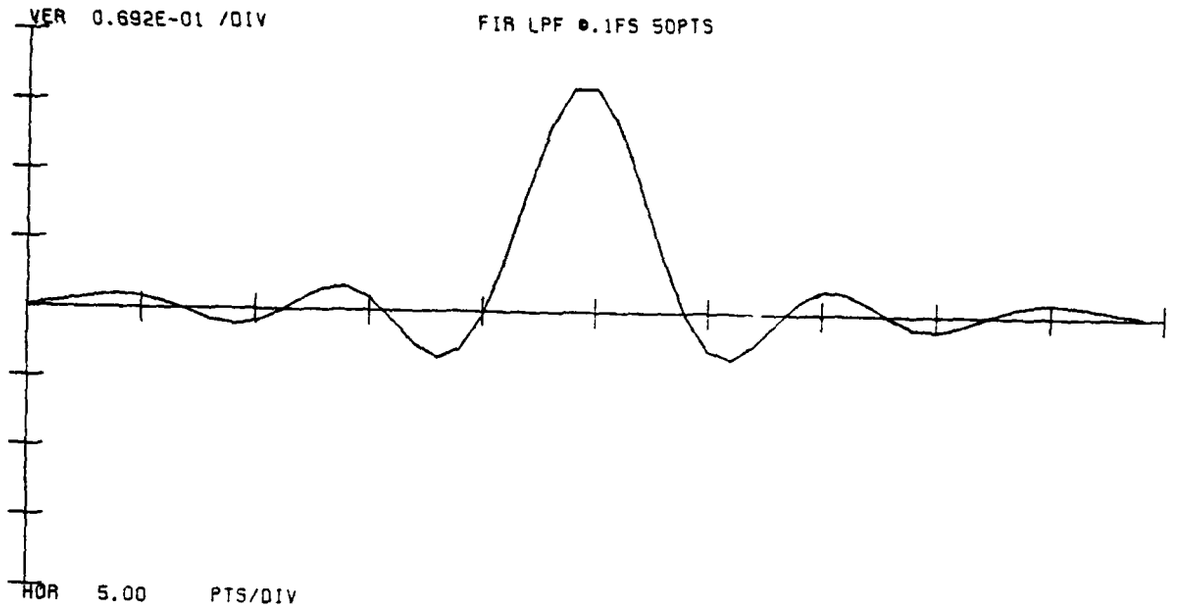

VER 200B/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.24c


VER 0.692E-01 /DIV
FIR LPF @.1FS 50PTS
HOR 5.00 PTS/DIV

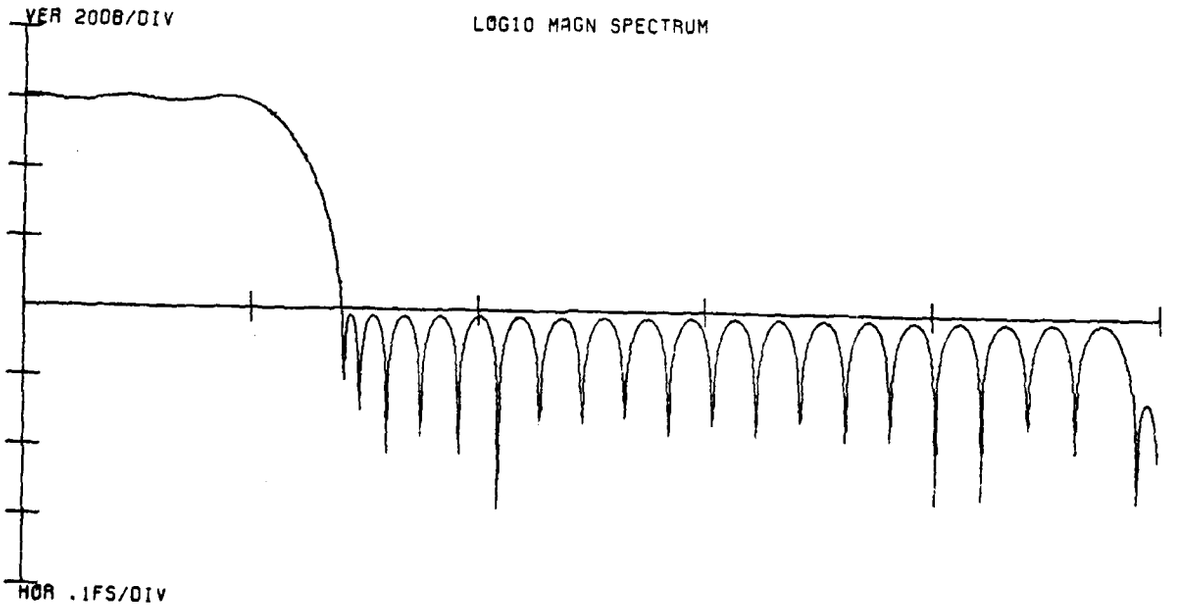
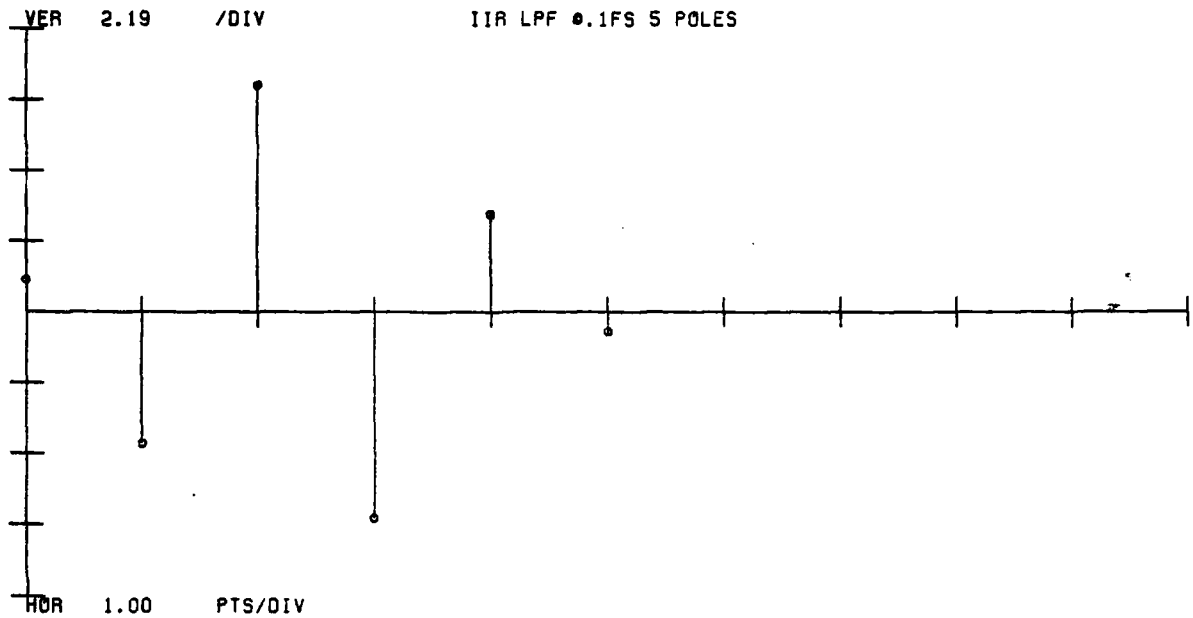

VER 200B/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.25


VER 2.19 /DIV
IIR LPF @.1FS 5 POLES
HOR 1.00 PTS/DIV

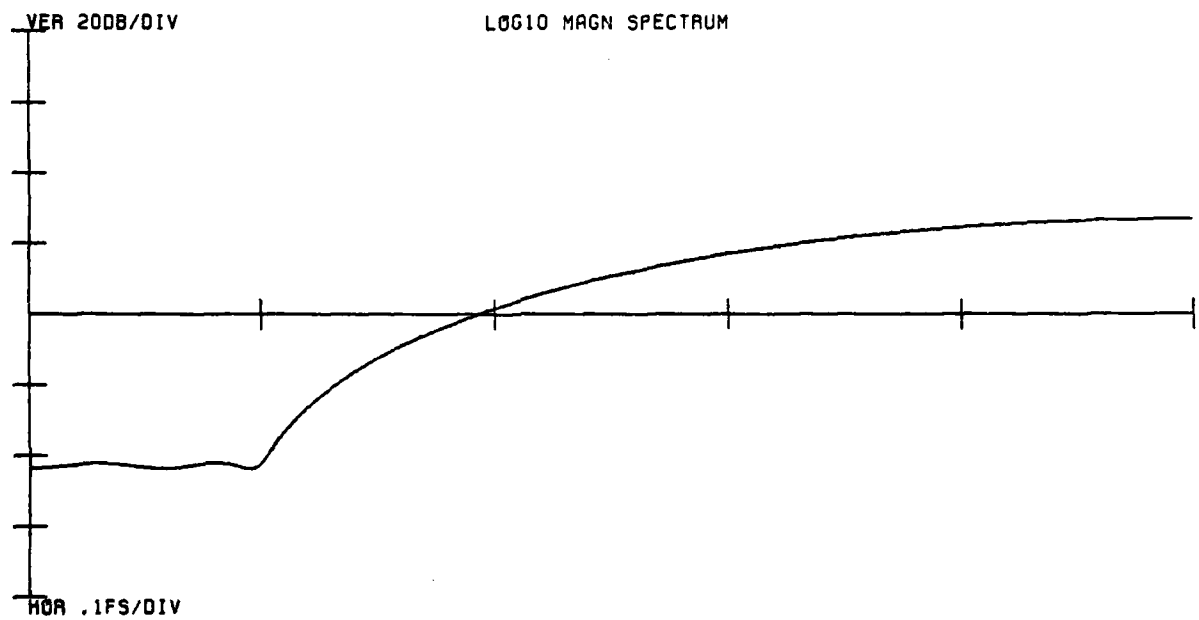

VER 20DB/DIV
LOG10 MAGN SPECTRUM
HOR .1FS/DIV

Figure 3.26

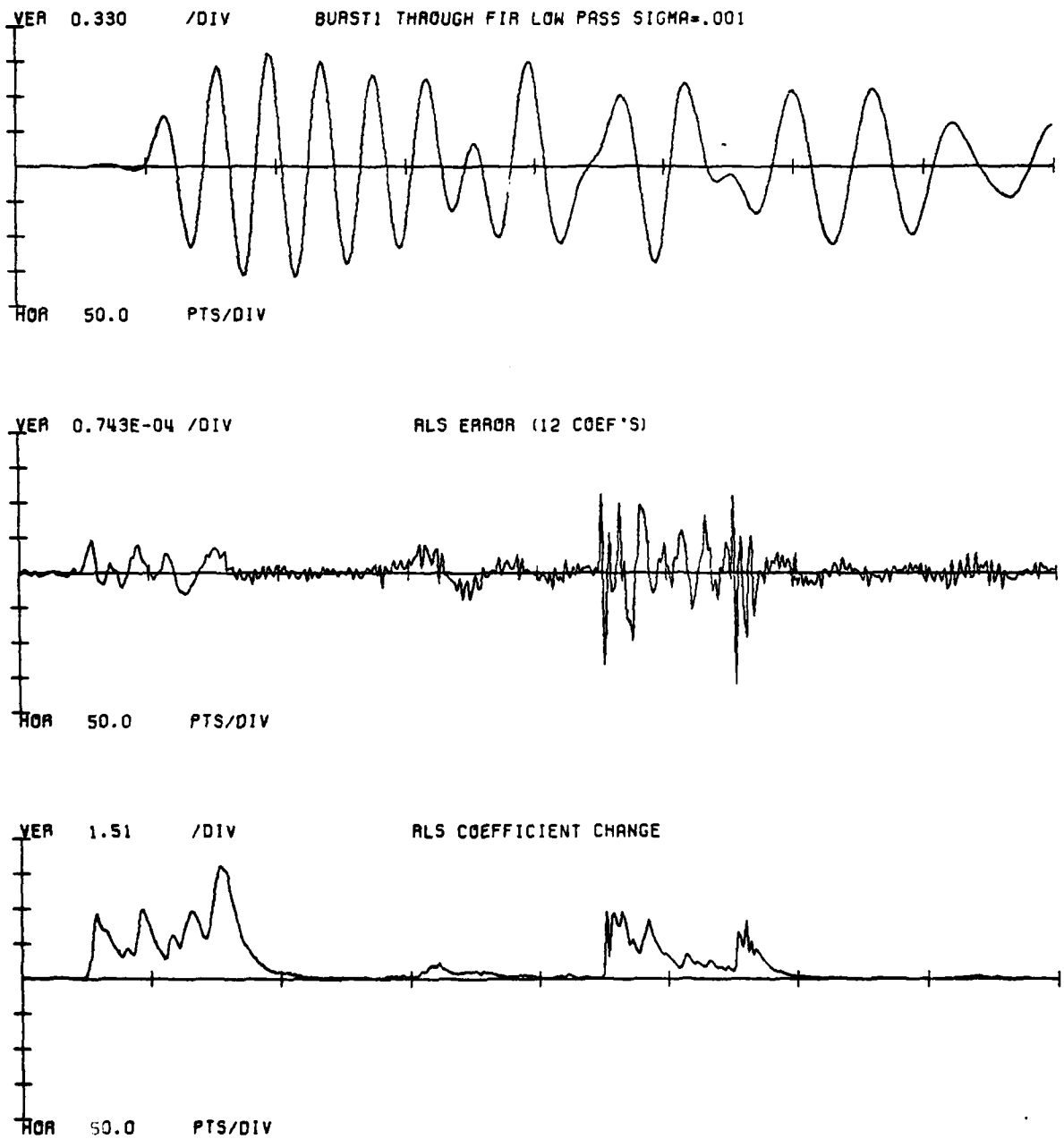

VER 0.330 /DIV
BURST1 THROUGH FIR LOW PASS SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.743E-04 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 1.51 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.27a

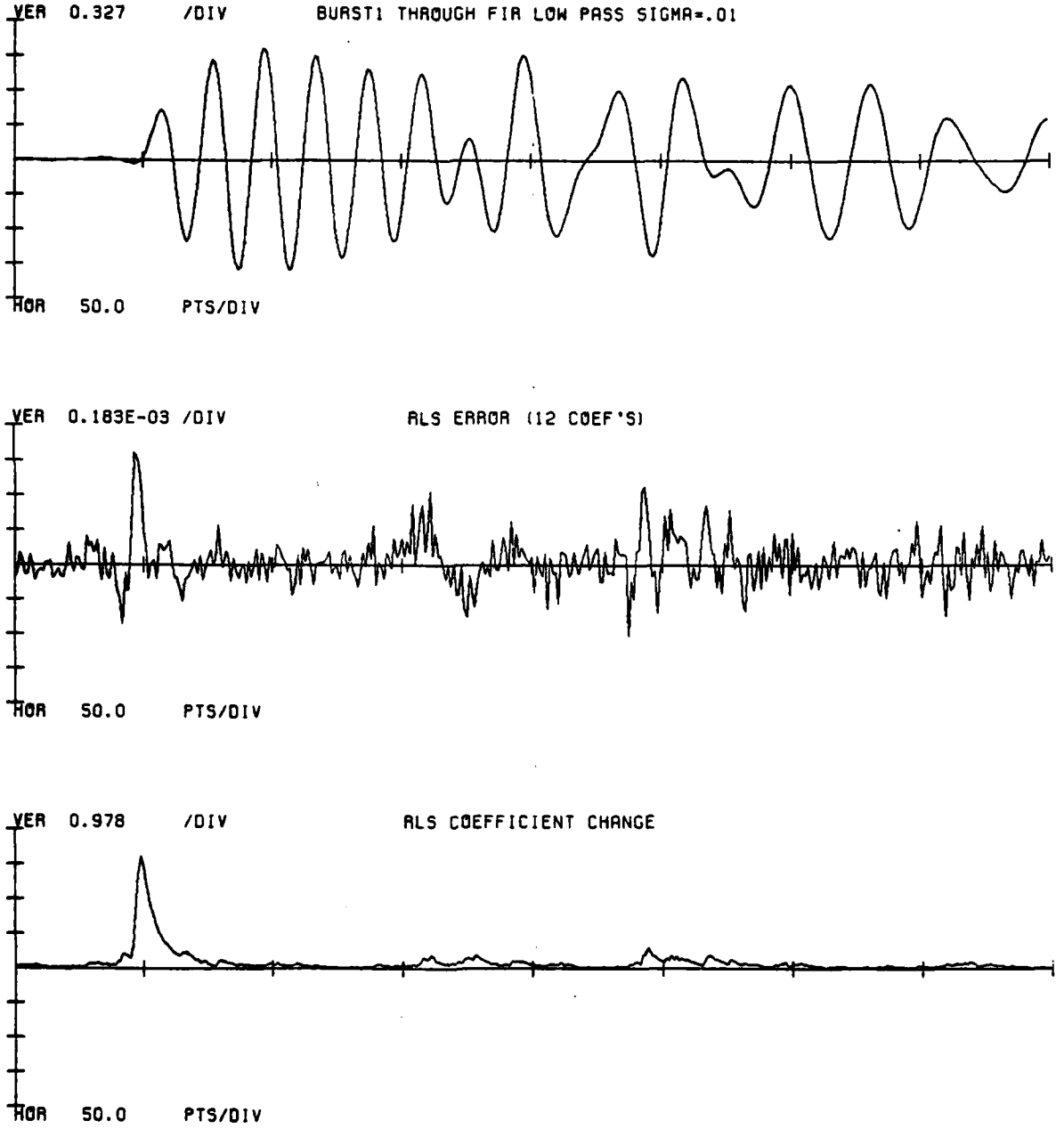

VER 0.327 /DIV
BURST1 THROUGH FIR LOW PASS SIGMA=.01
HOR 50.0 PTS/DIV
VER 0.183E-03 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.978 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.27b

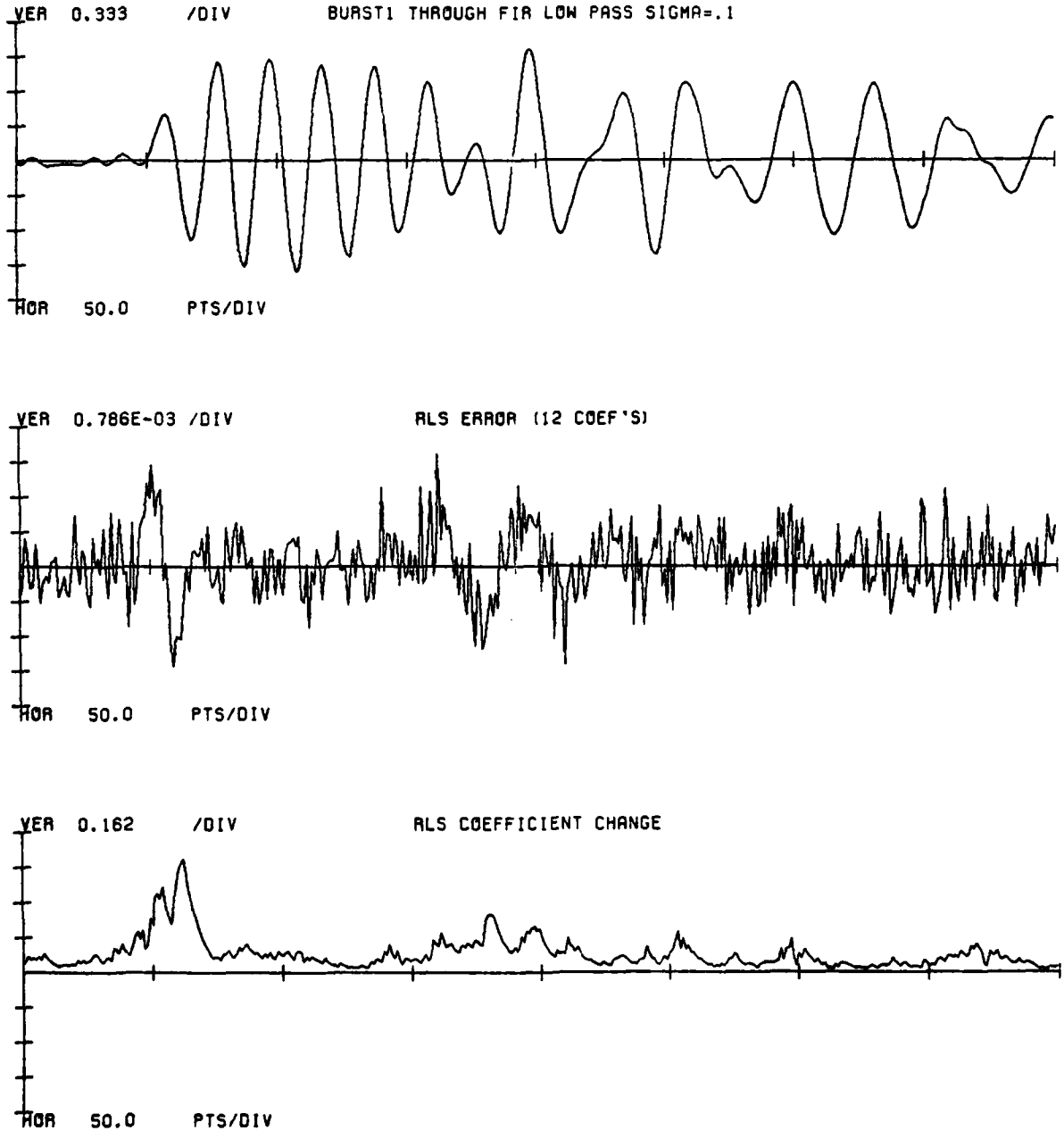

VER 0.333 /DIV
BURST1 THROUGH FIR LOW PASS SIGMA=.1
HOR 50.0 PTS/DIV
VER 0.786E-03 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.162 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.27c

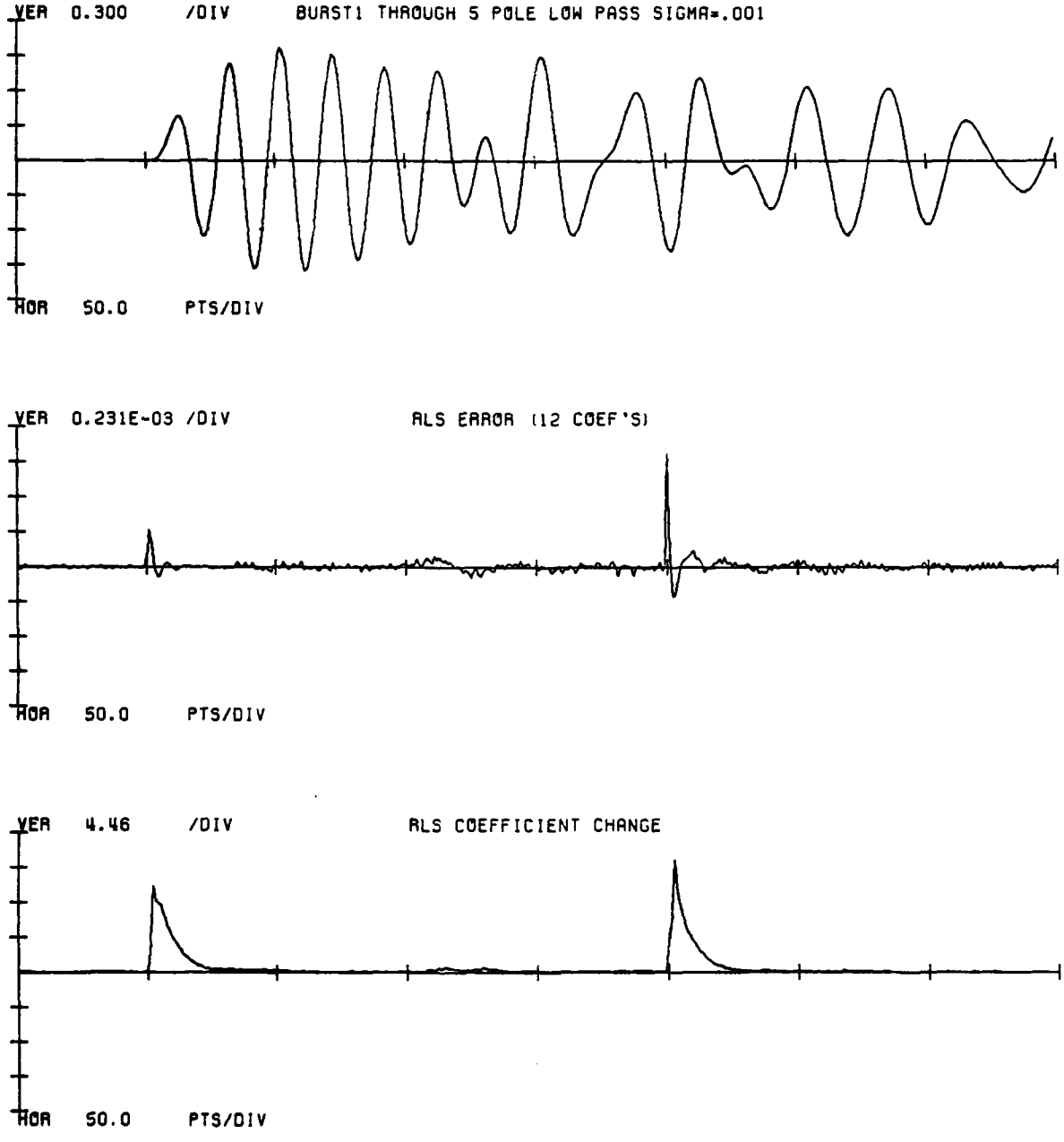
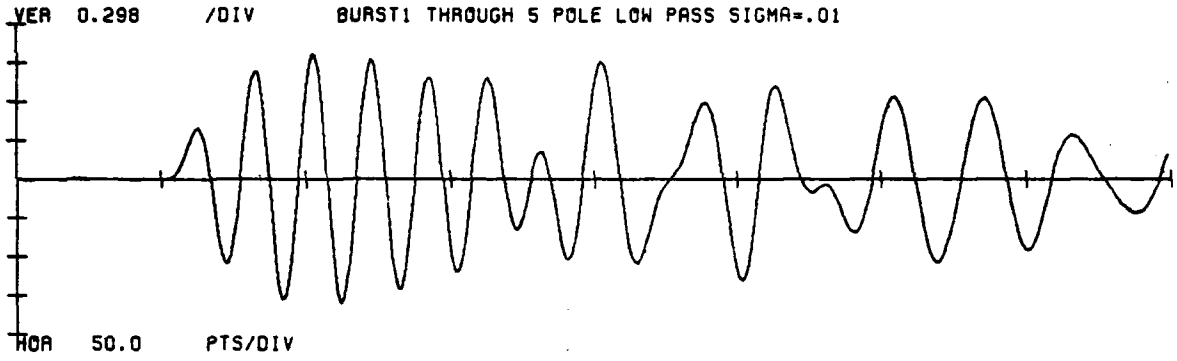
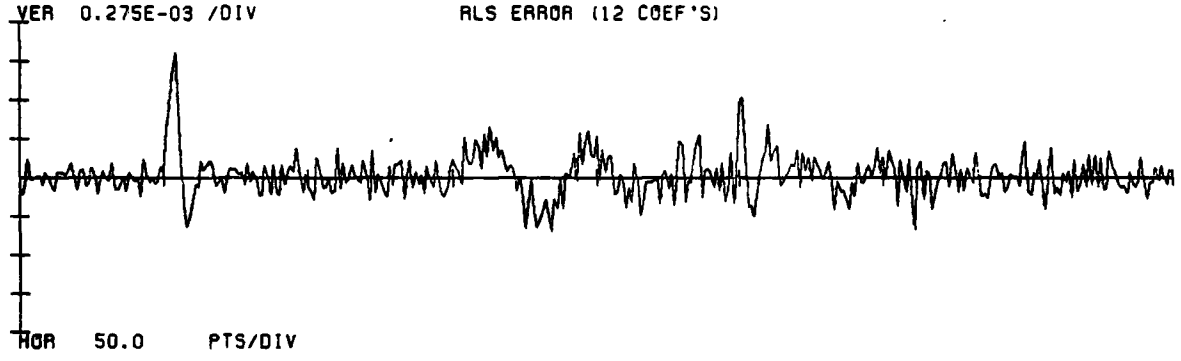

VER 0.300 /DIV
BURST1 THROUGH 5 POLE LOW PASS SIGMA=.001
HOR 50.0 PTS/DIV
VER 0.231E-03 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 4.46 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.28a


VER 0.298 /DIV
BURST1 THROUGH 5 POLE LOW PASS SIGMA=.01
HOR 50.0 PTS/DIV


VER 0.275E-03 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV

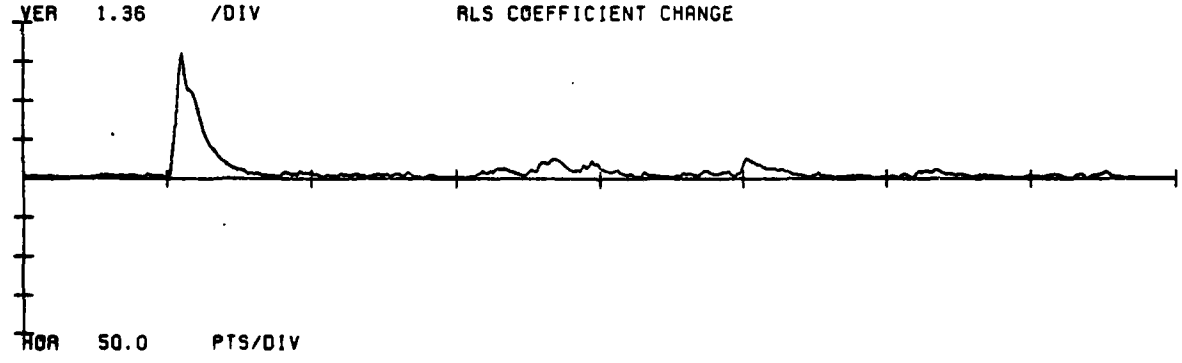

VER 1.36 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.28b

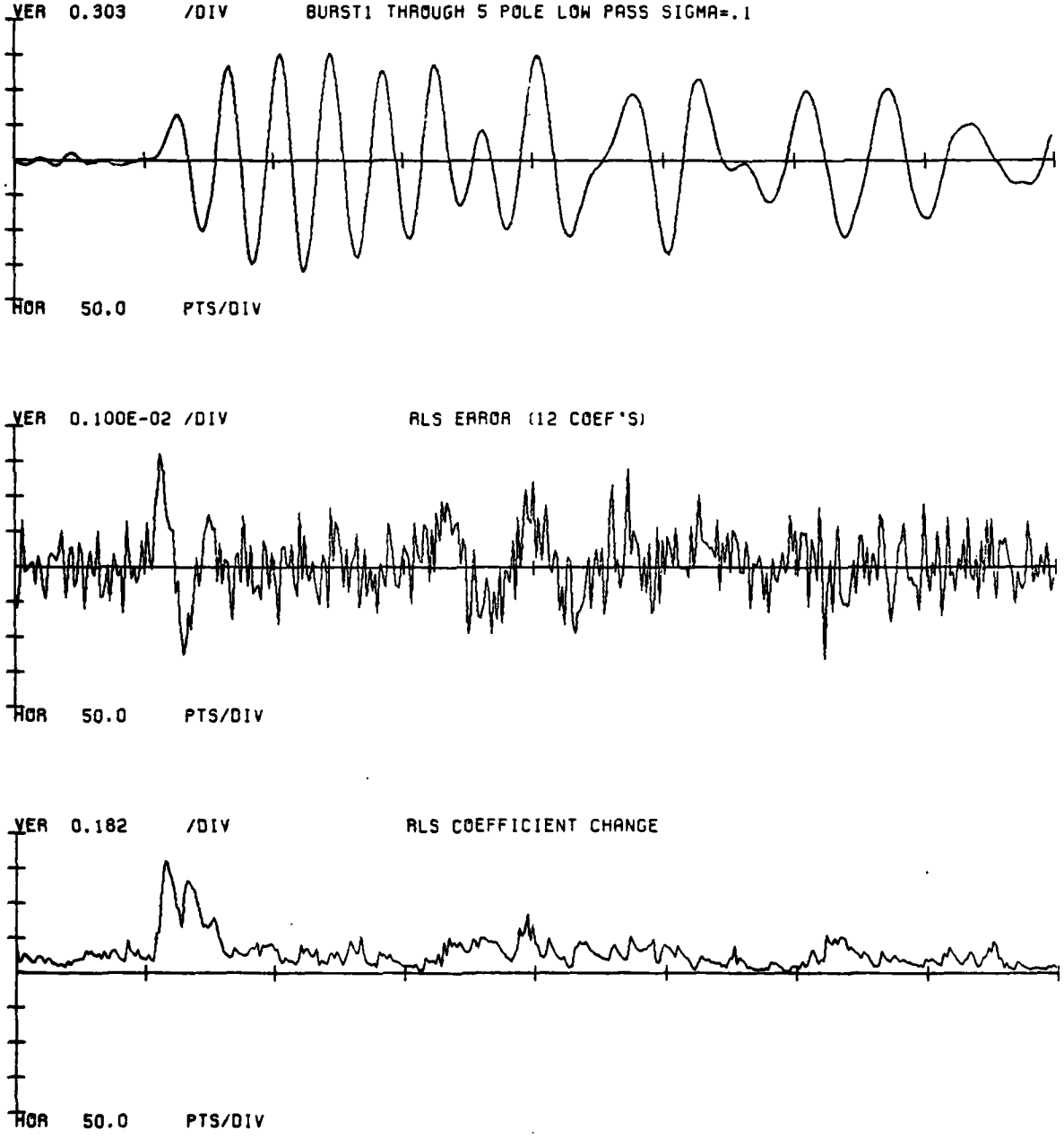
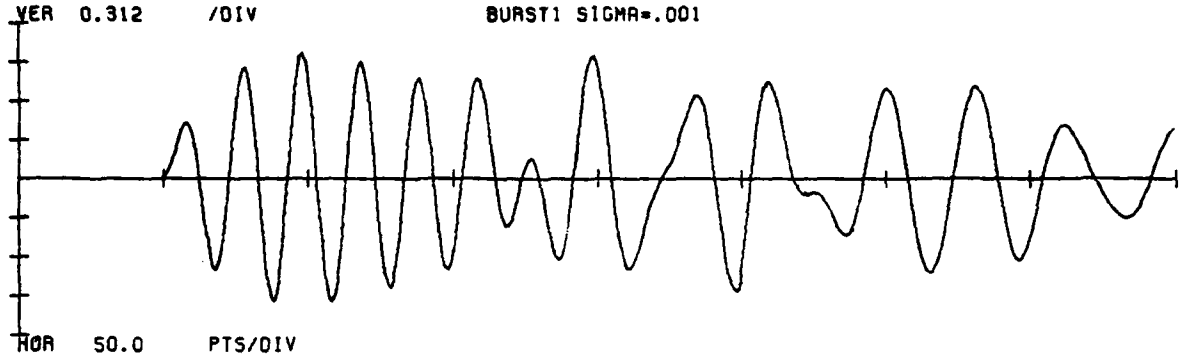
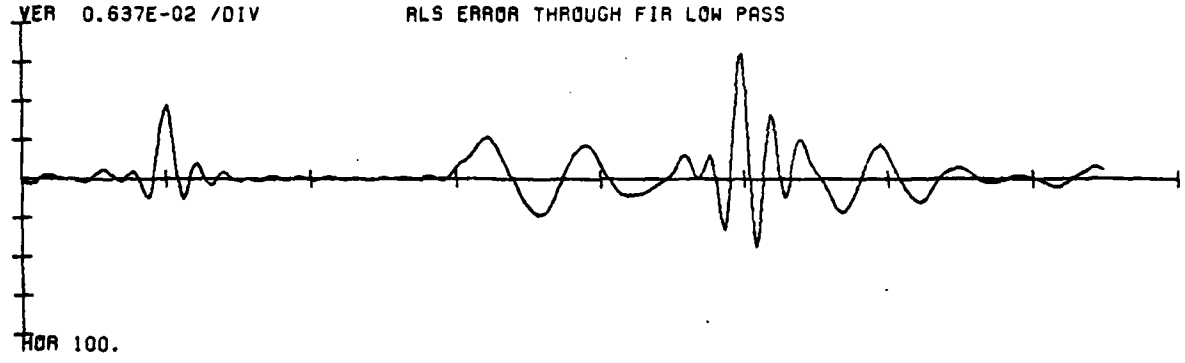

VER 0.303 /DIV
BURST1 THROUGH 5 POLE LOW PASS SIGMA=.1
HOR 50.0 PTS/DIV
VER 0.100E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 50.0 PTS/DIV
VER 0.182 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

FIgure 3.28c


VER 0.312 /DIV
BURST1 SIGMA=.001
HOR 50.0 PTS/DIV


VER 0.637E-02 /DIV
RLS ERROR THROUGH FIR LOW PASS
HOR 100.

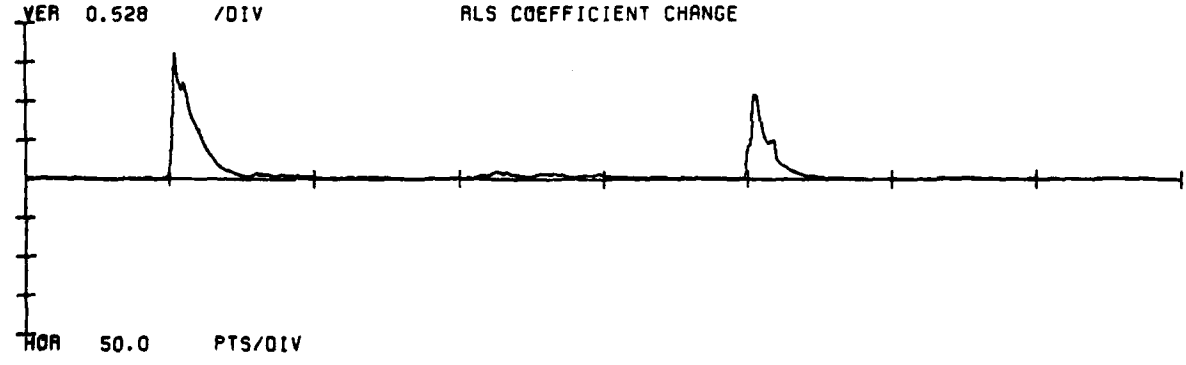
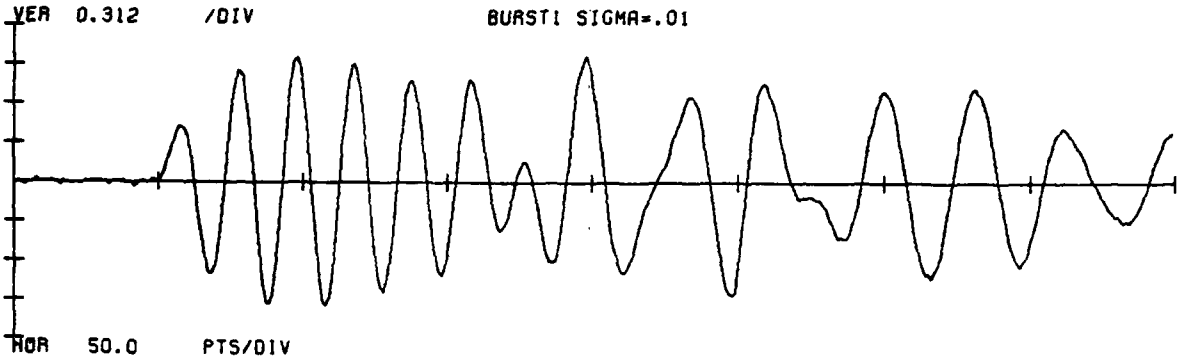
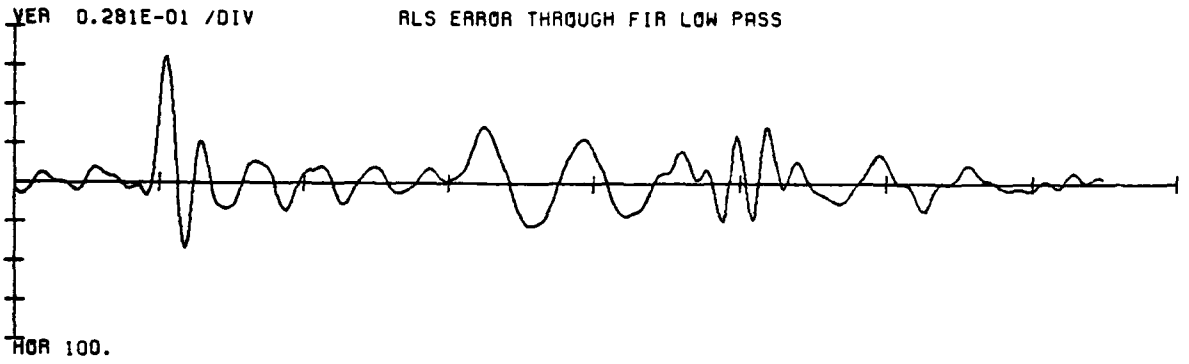

VER 0.528 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

FIgure 3.29a


VER 0.312 /DIV
BURST1 SIGMA=.01
HOR 50.0 PTS/DIV


VER 0.281E-01 /DIV
RLS ERROR THROUGH FIR LOW PASS
HOR 100.

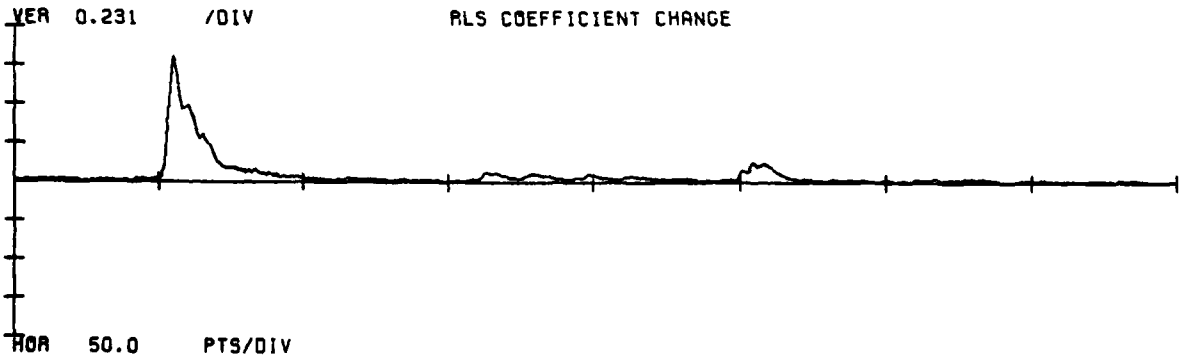
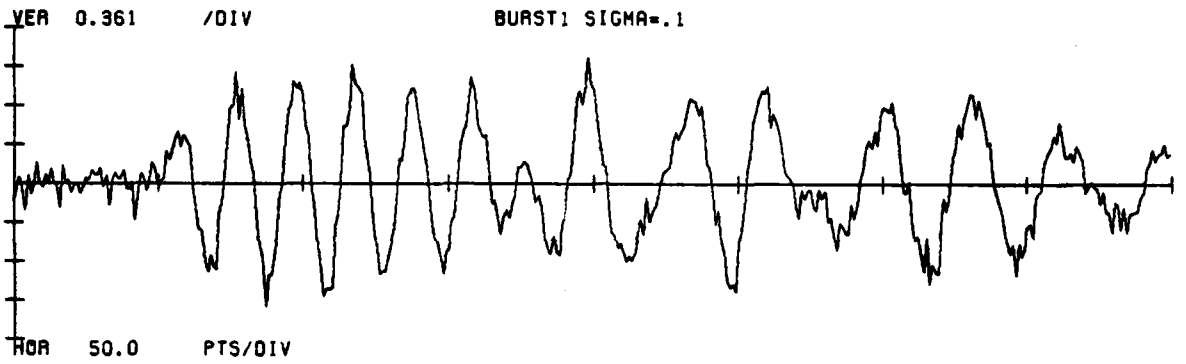
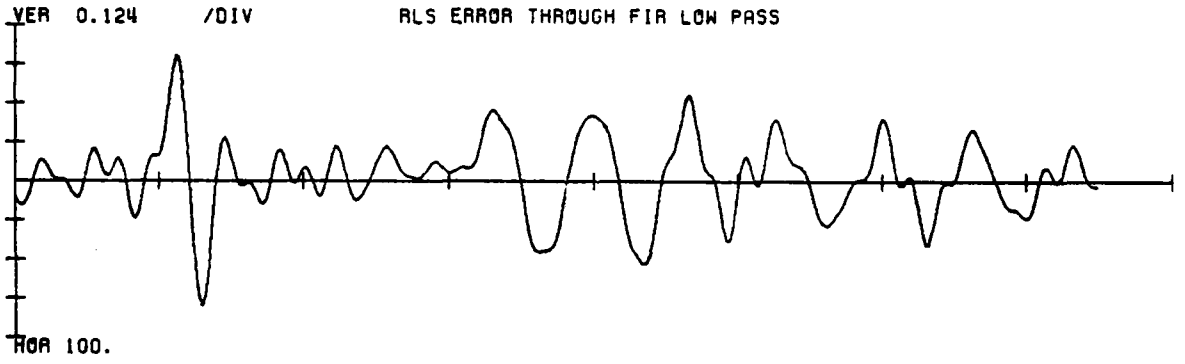

VER 0.231 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.29b


VER 0.361 /DIV
BURST1 SIGMA=.1
HOR 50.0 PTS/DIV


VER 0.124 /DIV
RLS ERROR THROUGH FIR LOW PASS
HOR 100.

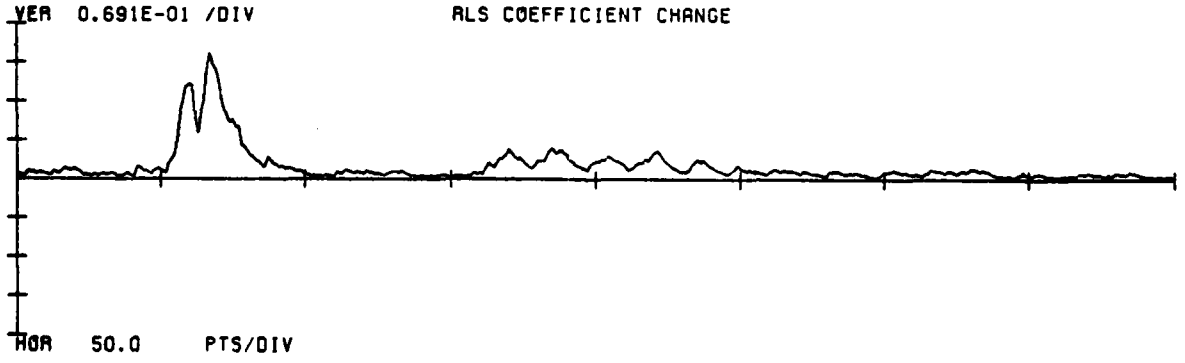

VER 0.691E-01 /DIV
RLS COEFFICIENT CHANGE
HOR 50.0 PTS/DIV

Figure 3.29c

3.5 Decimation

The band-limited nature of the burst1 signal suggests another possible approach to enhancement of the event compressed signals. The regions of the spectrum which have low energy (i.e. $.1f_{sample}$ to $.5f_{sample}$) are amplified, compared to the high energy regions, in the process of linear prediction (due to the whitening mentioned in the last section). Decimating the data would reduce the relative size of the low energy region of the spectrum possibly reducing the impact that the noise in that region had on the event compressed signals and allowing the predictor to expend more "effort" predicting the events and less predicting the noise.

Figures 3.30a-c and 3.31a-c show 2/1 decimation of the Burst1 signal with and without all-pole prefiltering to reduce aliasing. Figures 3.32a-c and 3.33a-c show the same examples but with 4/1 decimation.[1]

Comparison of these figures with the undecimated examples (figs. 3.17 and 3.28) show that the coefficients change more at each event with increasing decimation ratios, but neither the unfiltered nor the filtered decimation methods offers substantial improvement in the quality of the data. In

---

1. All-pole filtering was chosen over FIR filtering because it smears the events less.

fact since the settling time of the predictor seems to be independent of the decimation ratio, and since decimation lowers the interevent spacing, interevent interference is more likely if the data has been decimated.

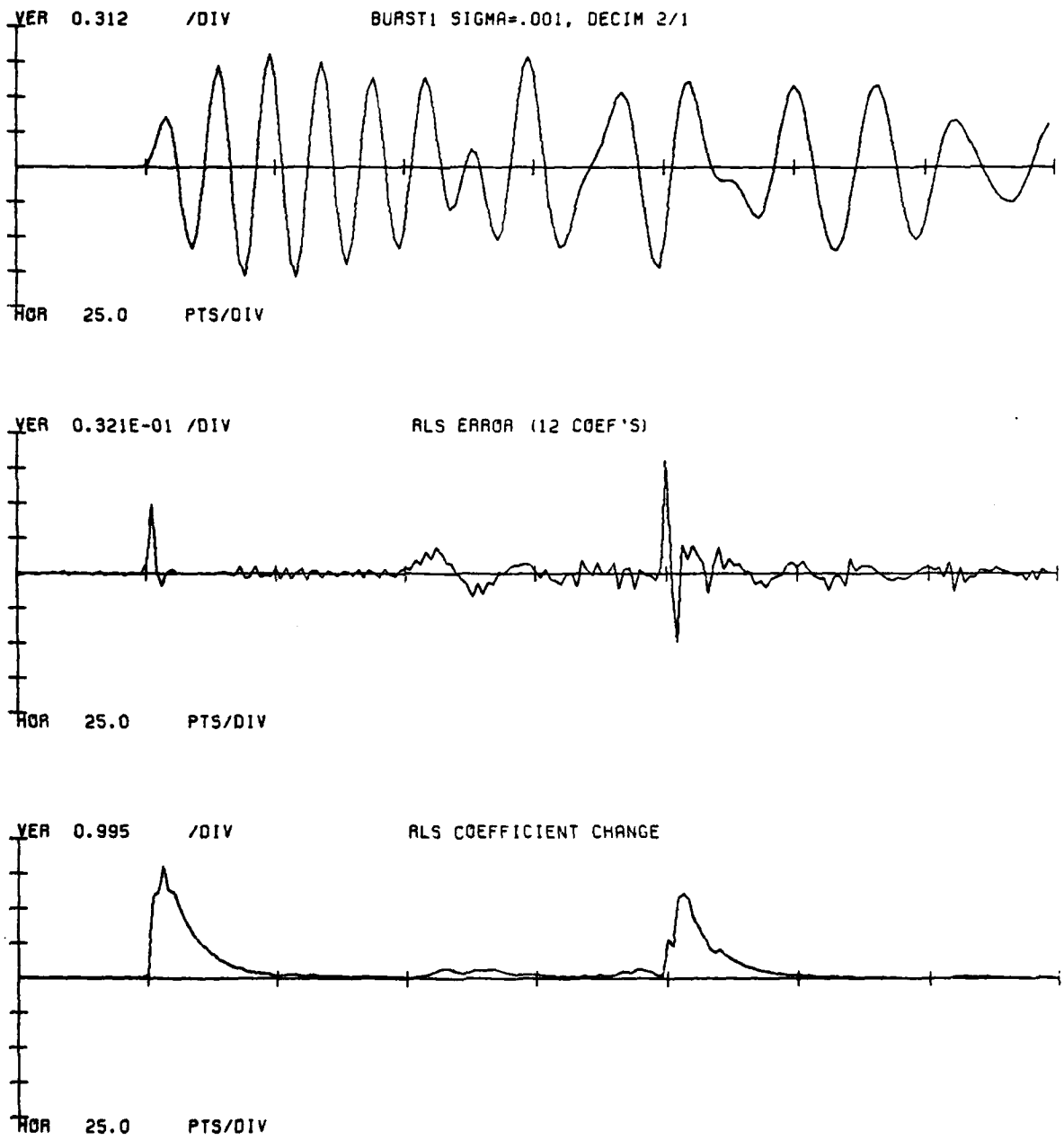

VER 0.312 /DIV
BURST1 SIGMA=.001, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.321E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 0.995 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.30a

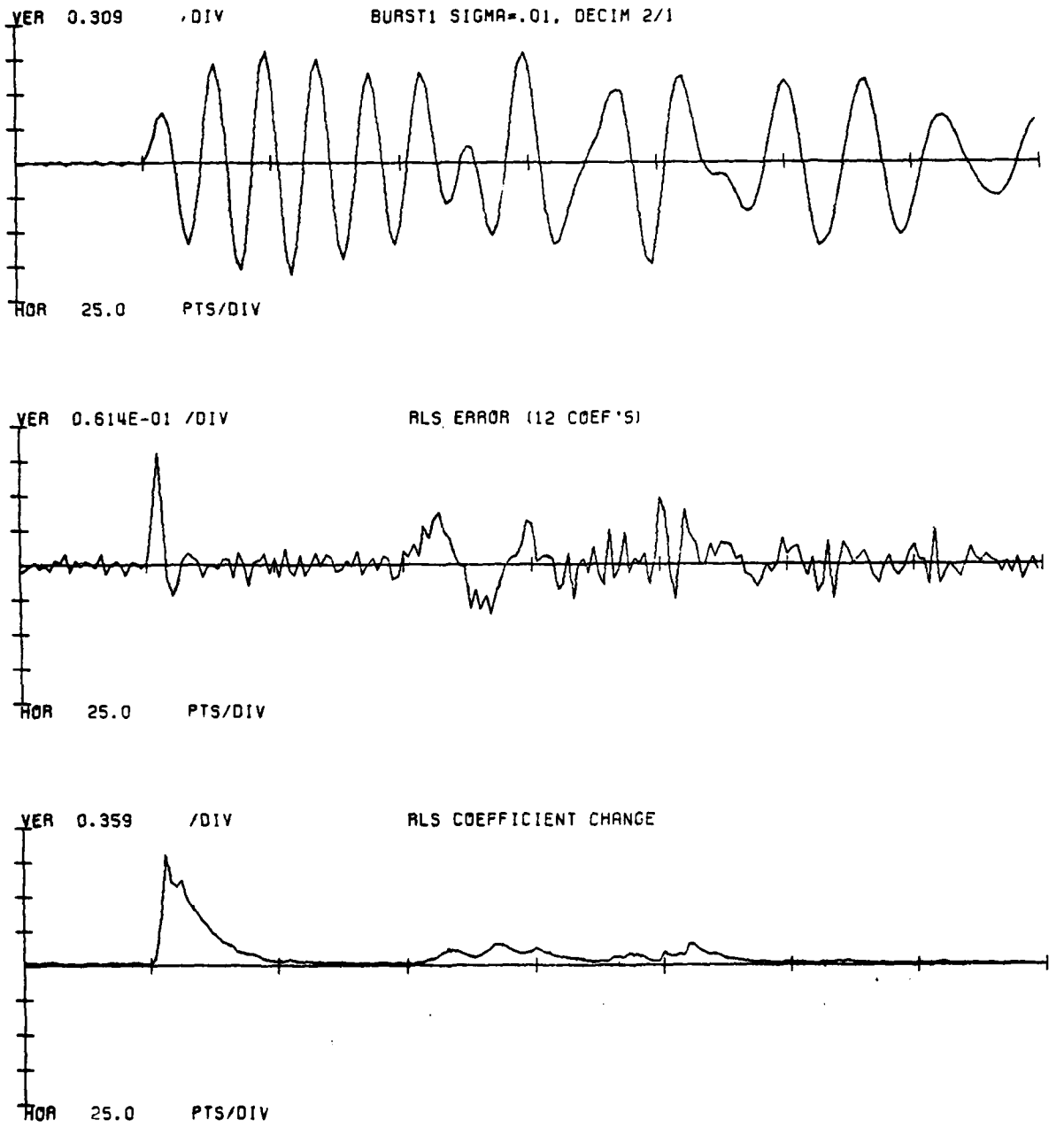

VER 0.309 /DIV
BURST1 SIGMA=.01, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.614E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 0.359 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.30b

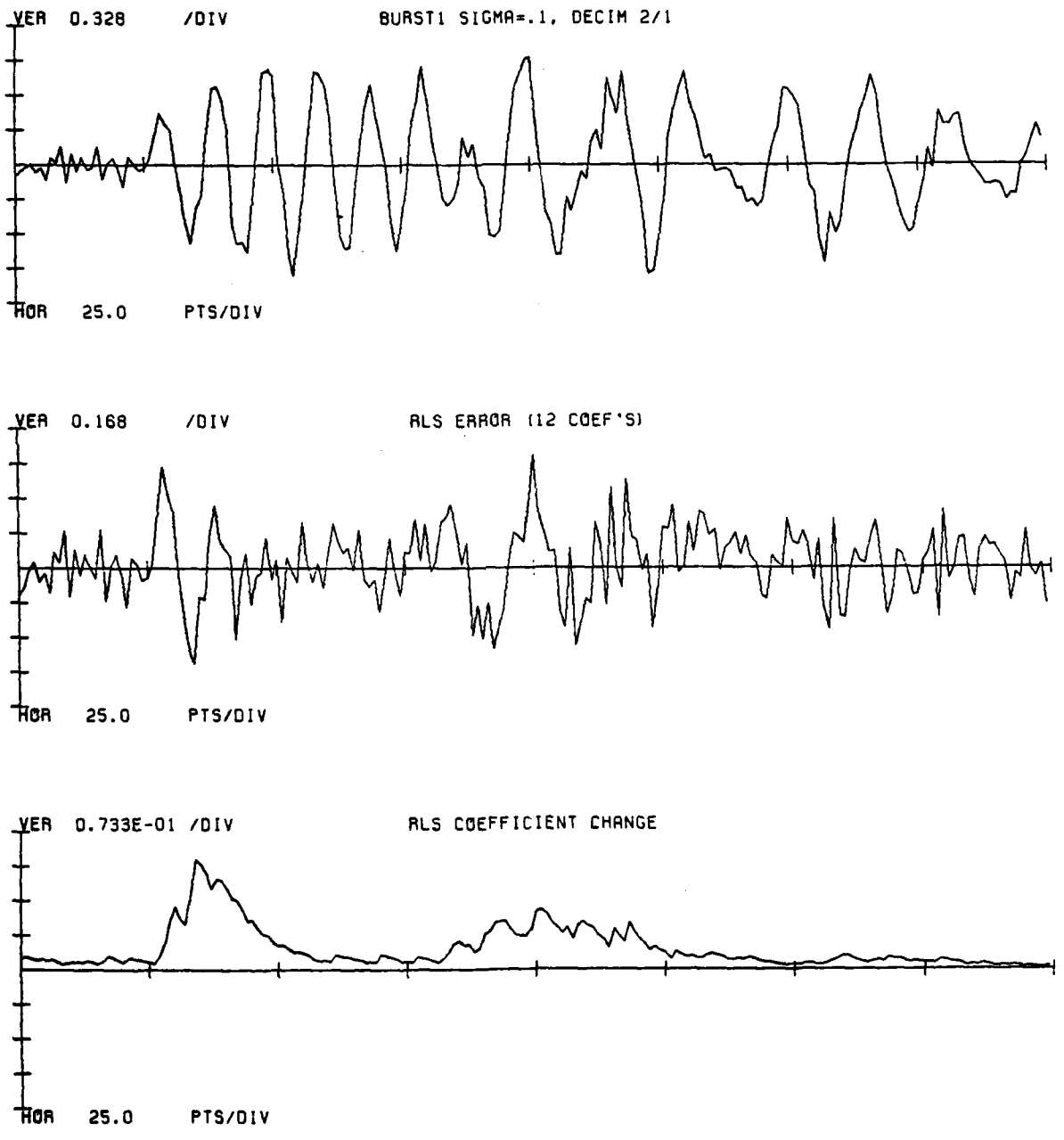

VER 0.328 /DIV
BURST1 SIGMA=.1, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.168 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 0.733E-01 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.30c

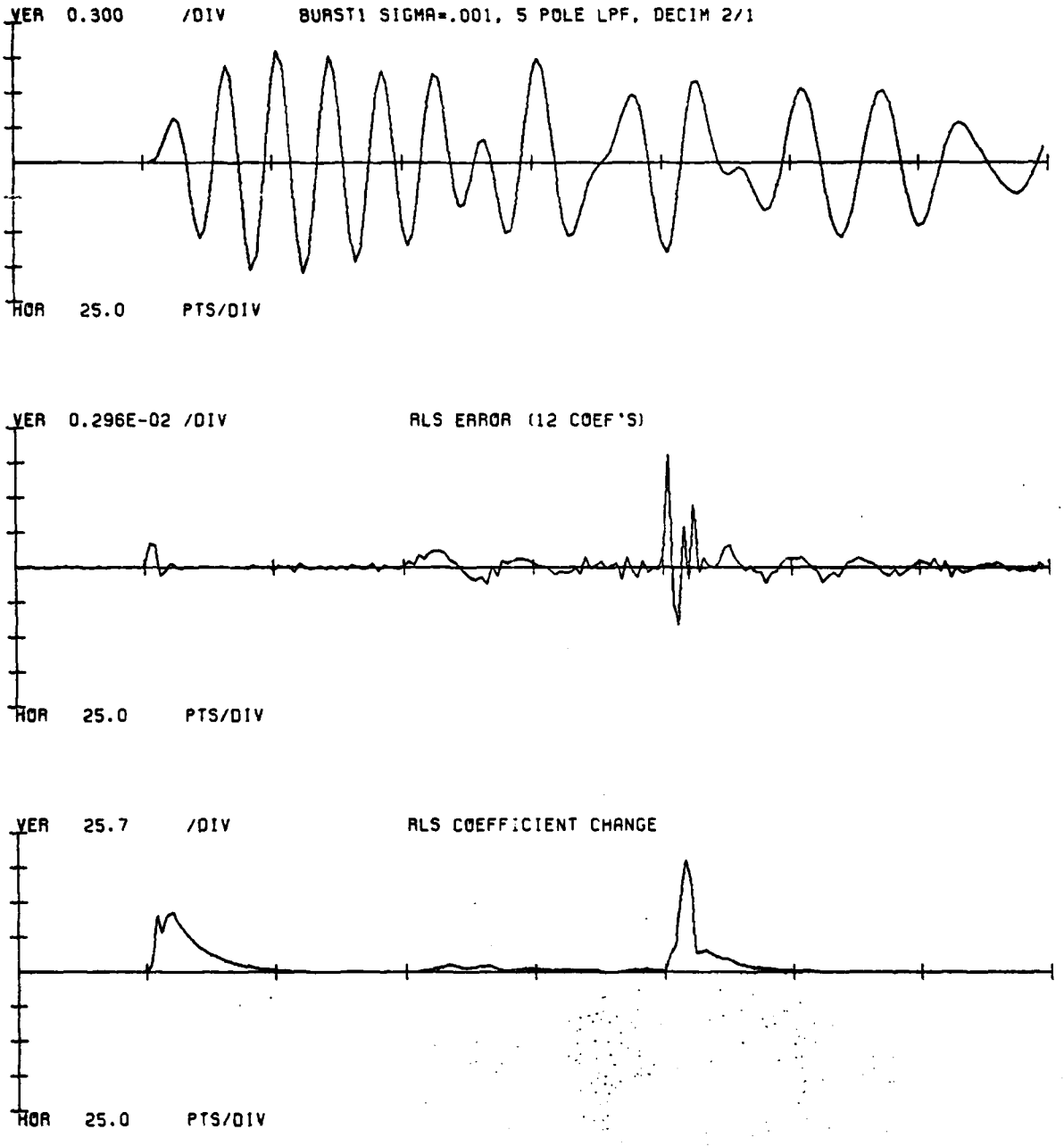

VER 0.300 /DIV
BURST1 SIGMA=.001, 5 POLE LPF, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.296E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 25.7 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.31a

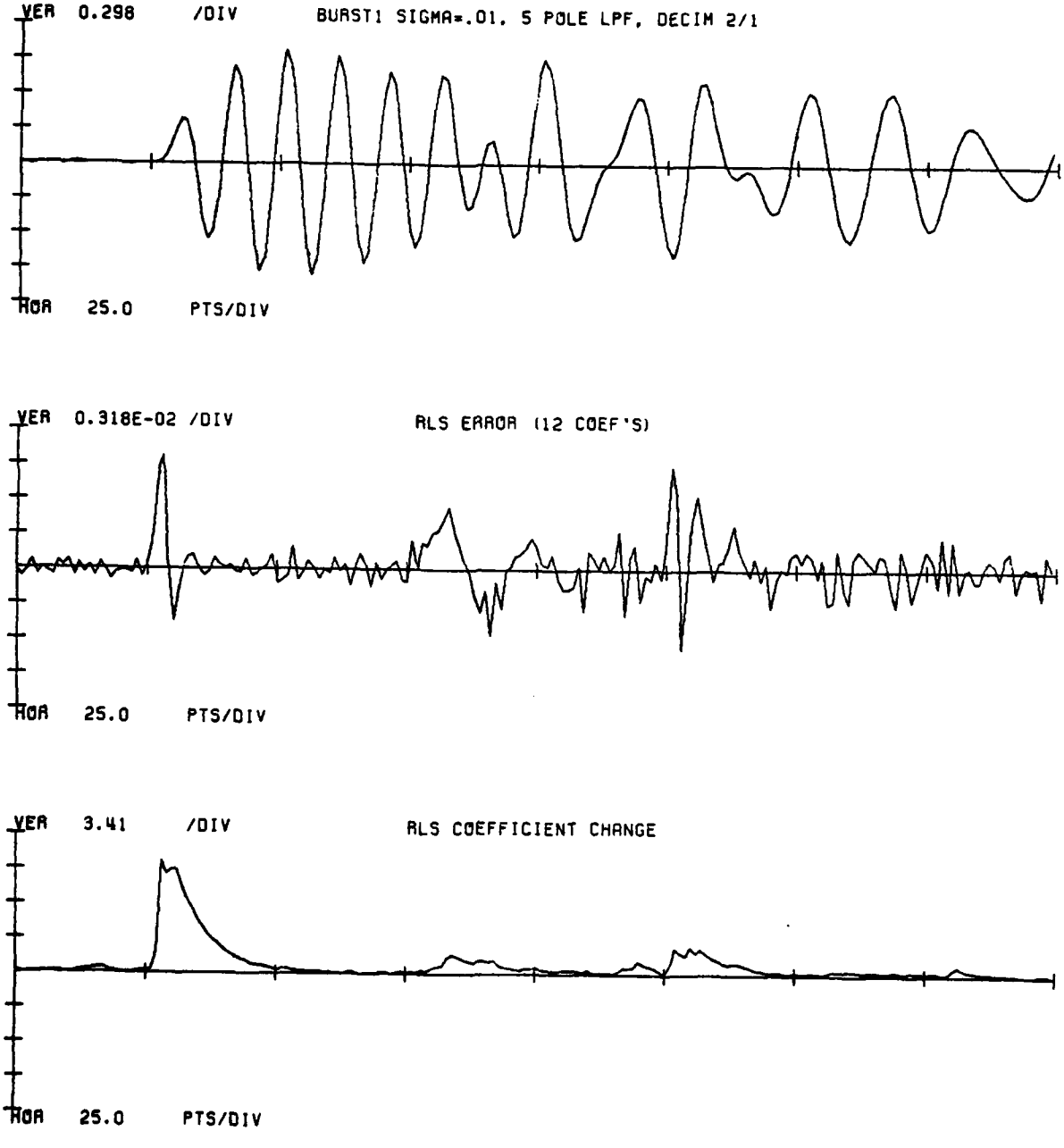

VER 0.298 /DIV
BURST1 SIGMA=.01, 5 POLE LPF, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.318E-02 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 3.41 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.31b

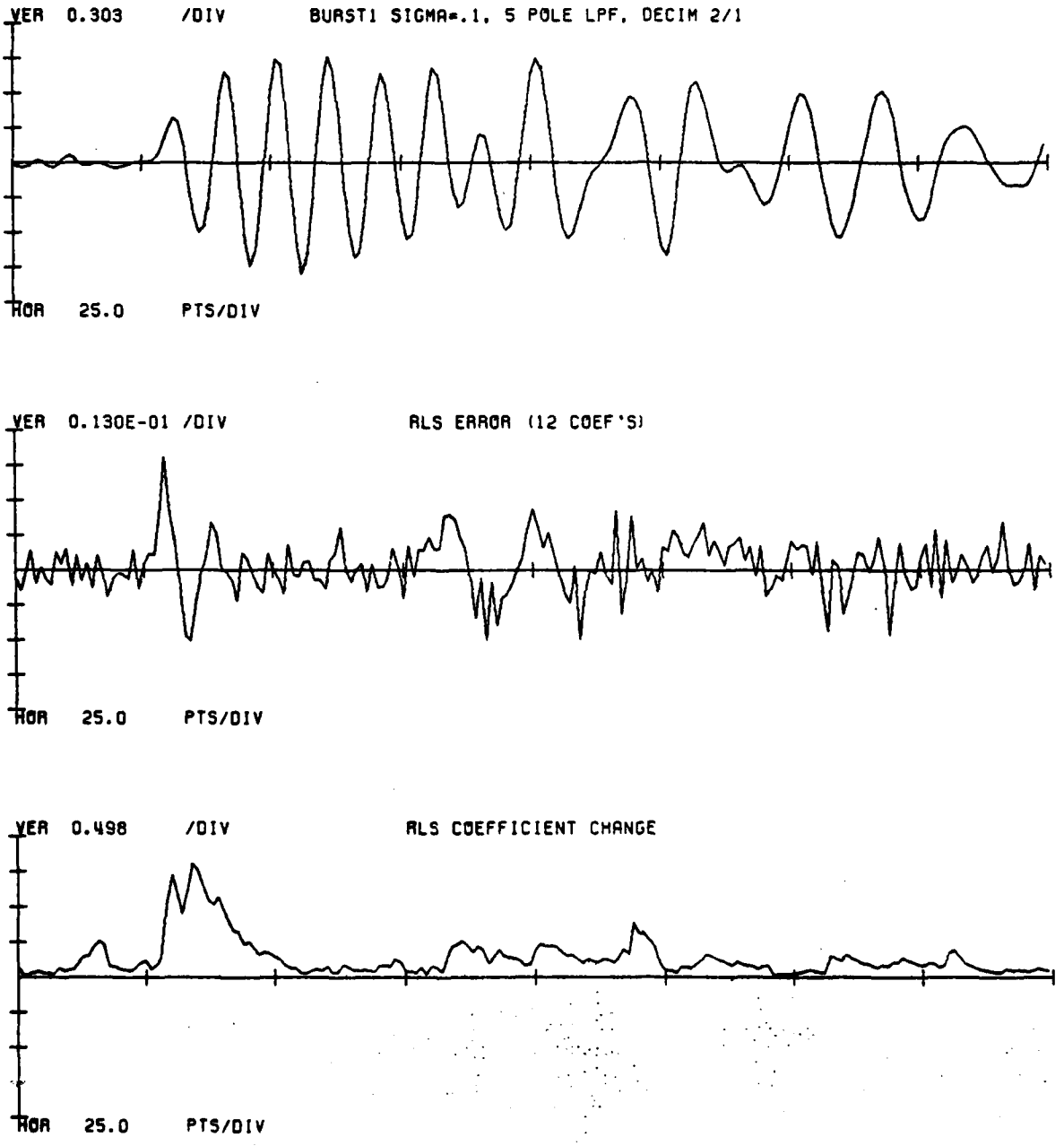

VER 0.303 /DIV
BURST1 SIGMA=.1, 5 POLE LPF, DECIM 2/1
HOR 25.0 PTS/DIV
VER 0.130E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 25.0 PTS/DIV
VER 0.498 /DIV
RLS COEFFICIENT CHANGE
HOR 25.0 PTS/DIV

Figure 3.31c

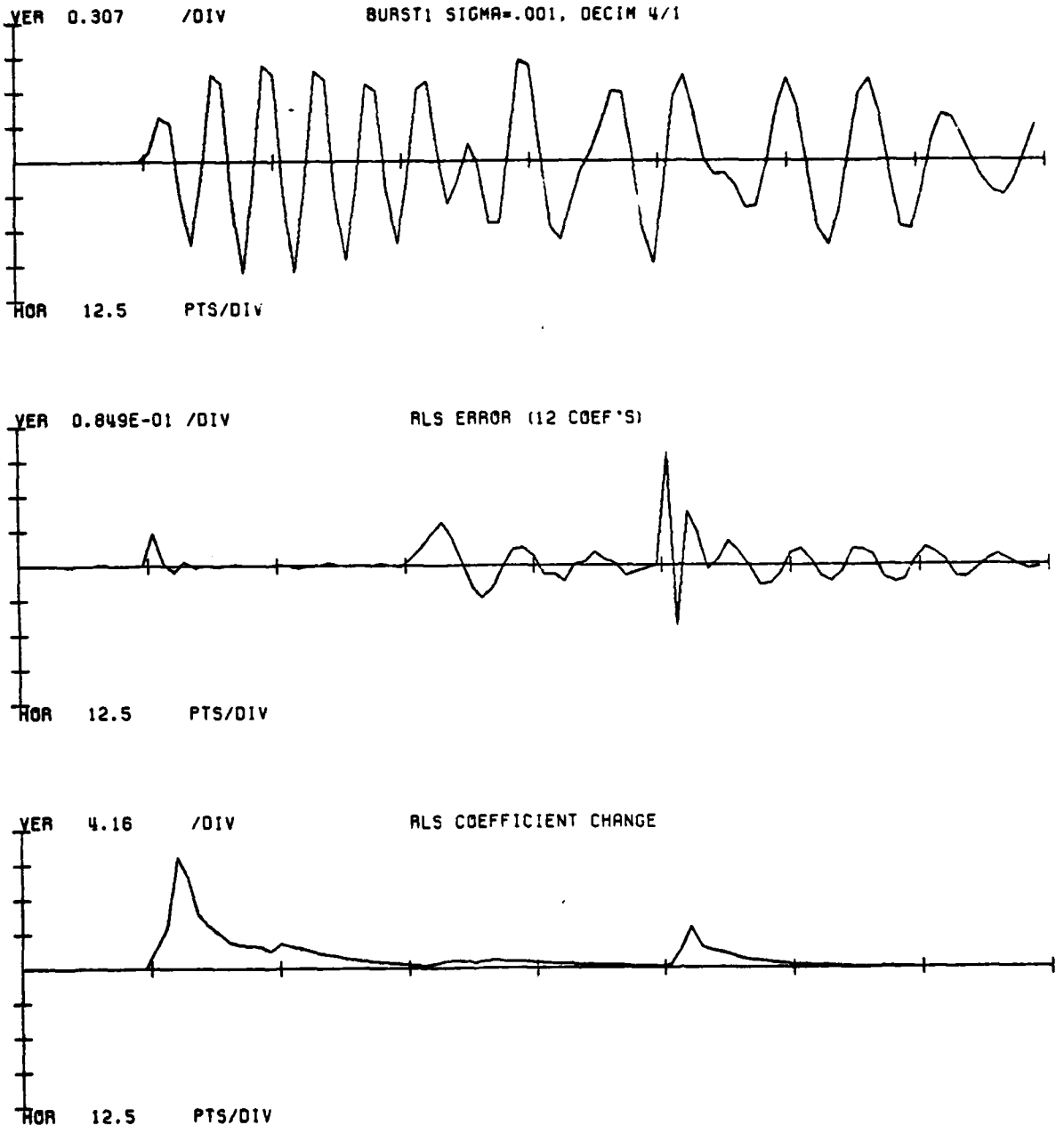

VER 0.307 /DIV
BURST1 SIGMA=.001, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.849E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 4.16 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.32a

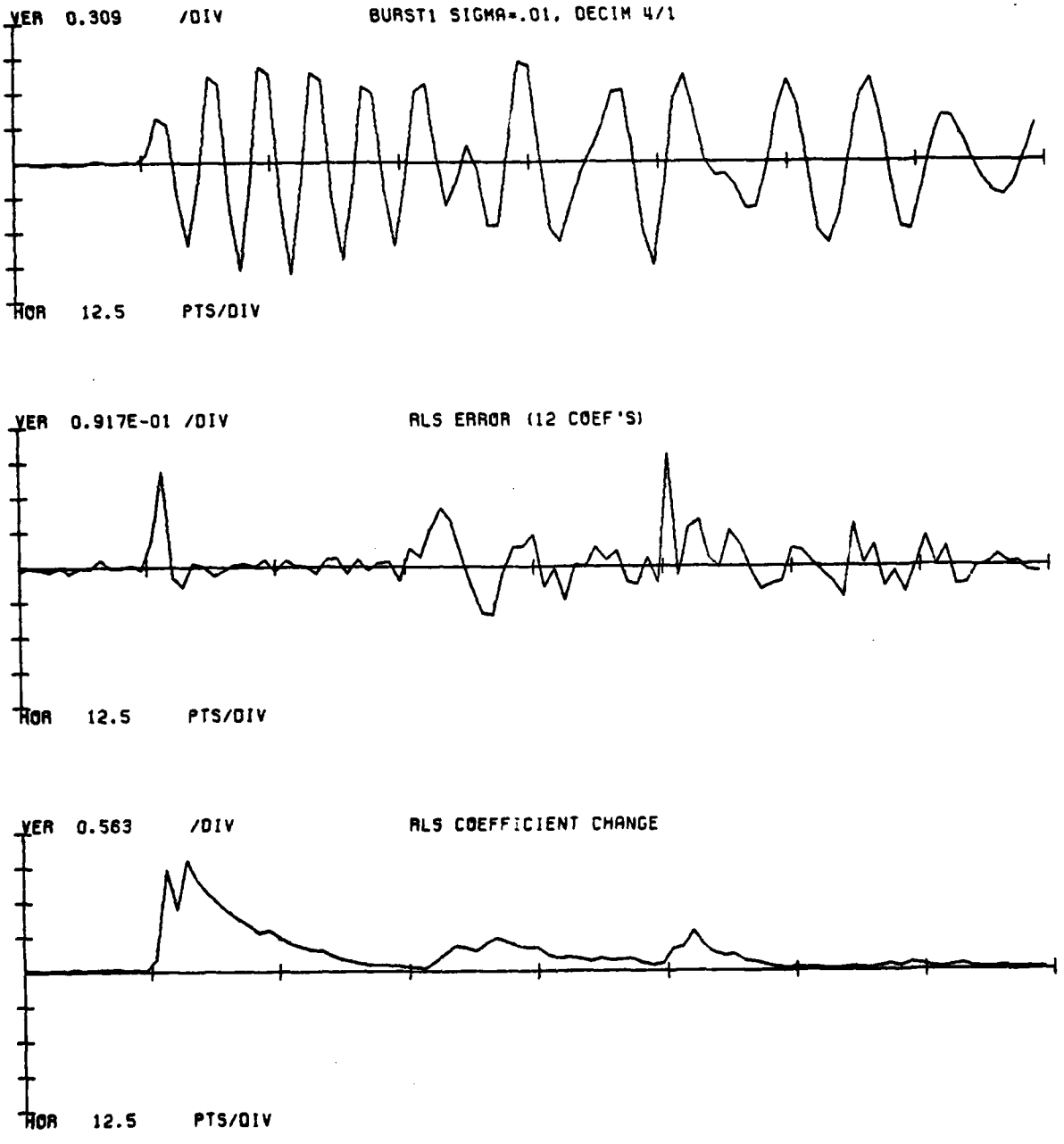

VER 0.309 /DIV
BURST1 SIGMA=.01, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.917E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 0.563 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.32b

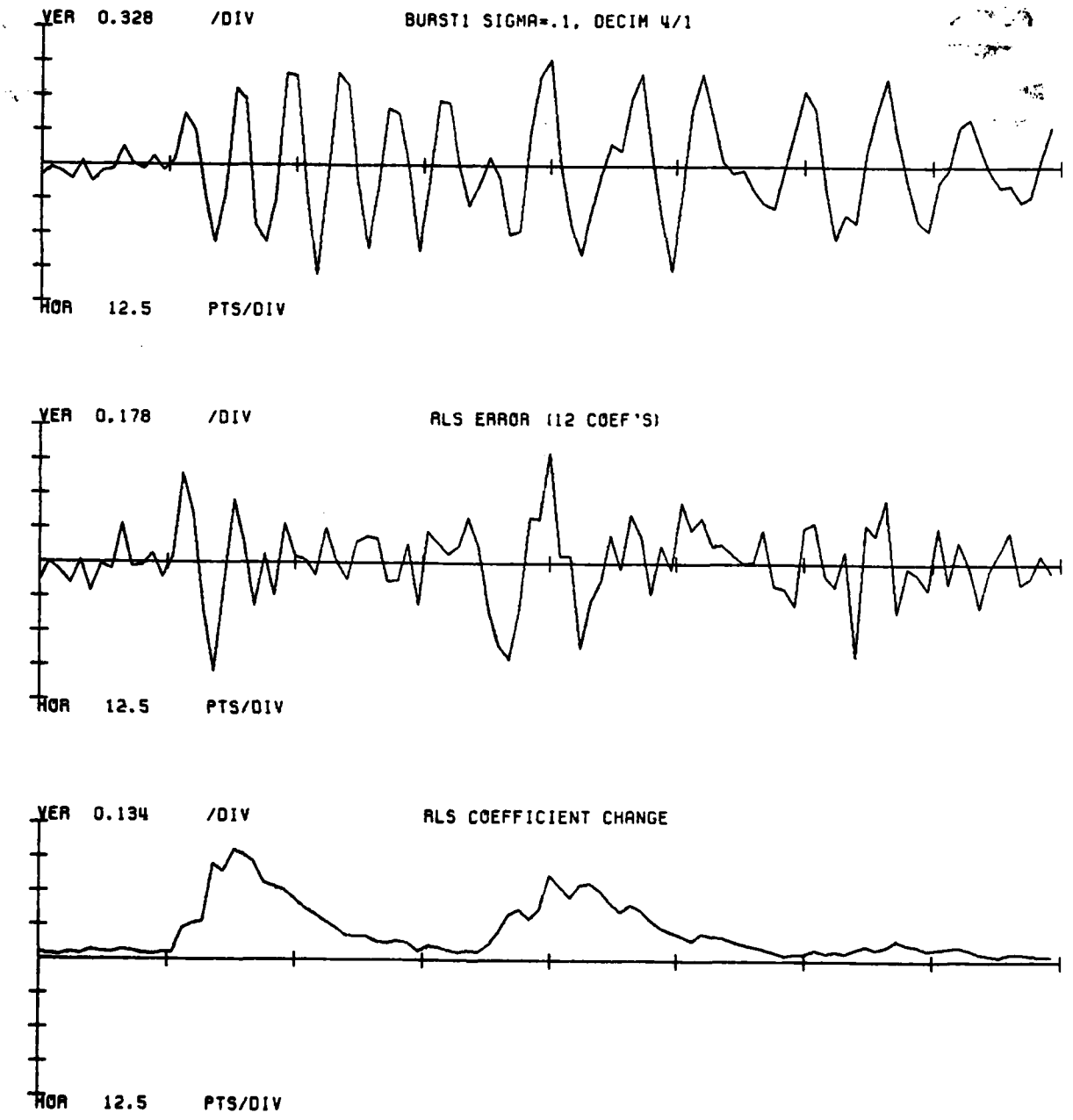

VER 0.328 /DIV
BURST1 SIGMA=.1, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.178 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 0.134 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.32c

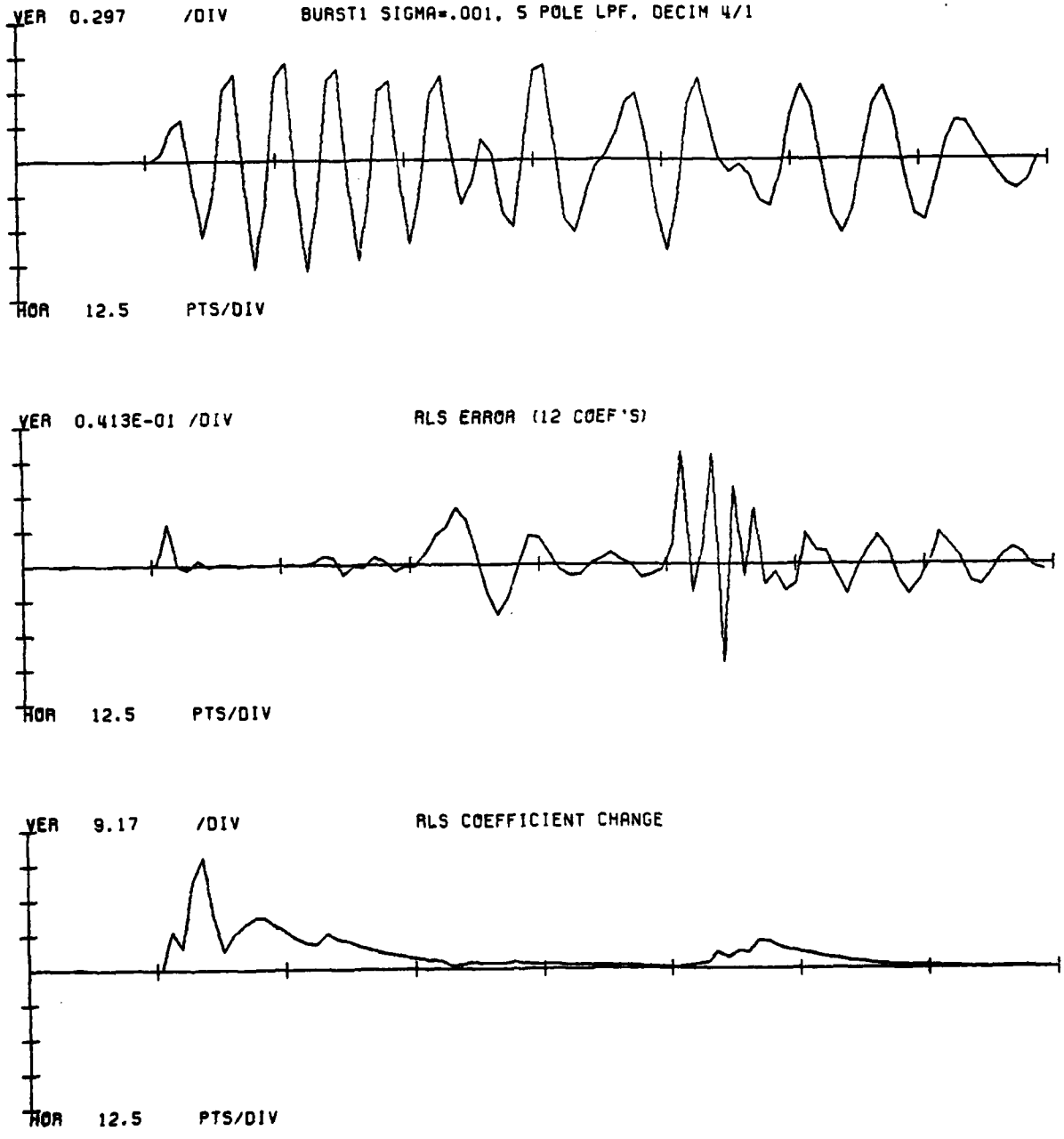

VER 0.297 /DIV
BURST1 SIGMA=.001, 5 POLE LPF, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.413E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 9.17 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.33a

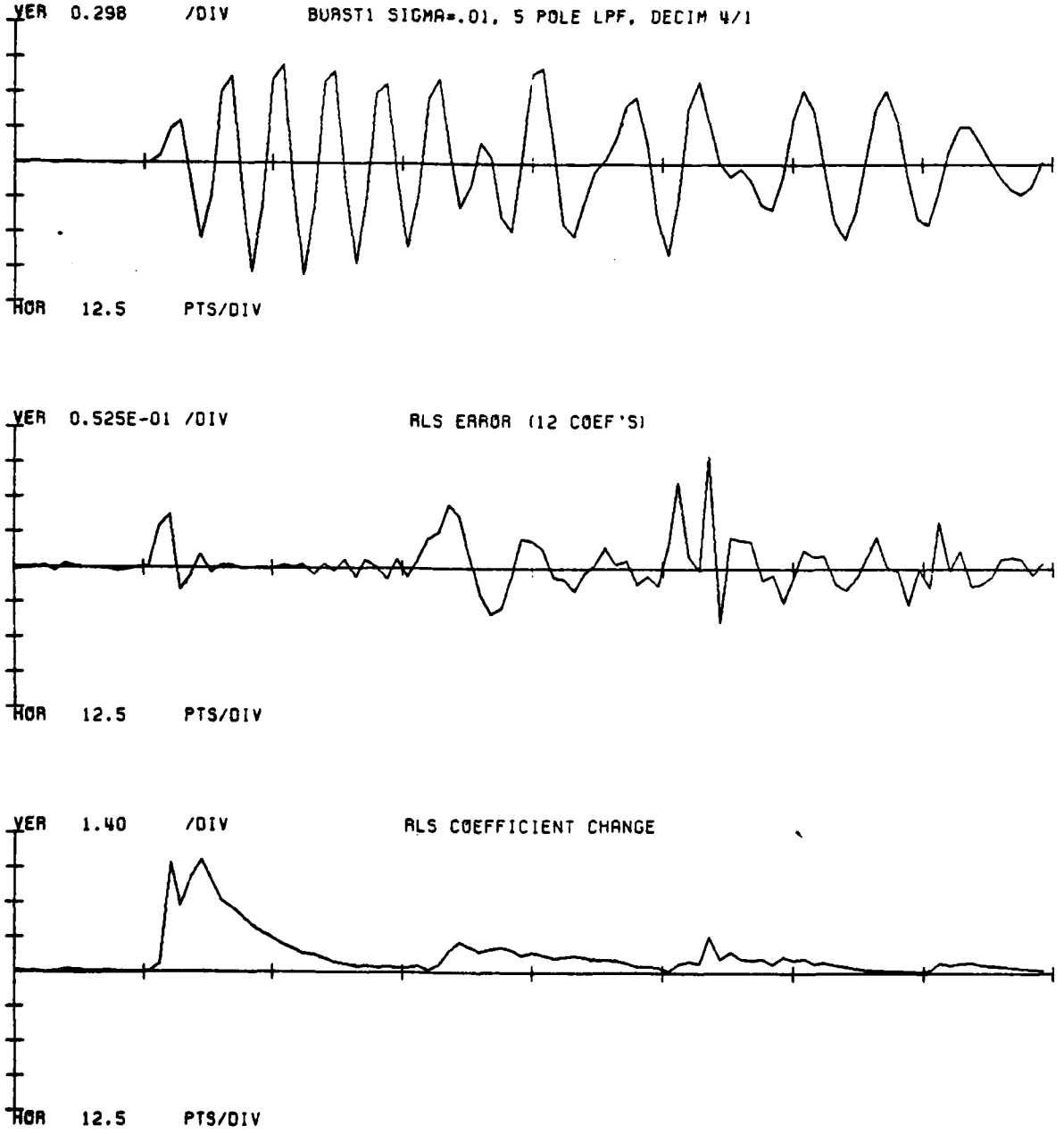

VER 0.298 /DIV
BURST1 SIGMA=.01, 5 POLE LPF, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.525E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 1.40 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.33b

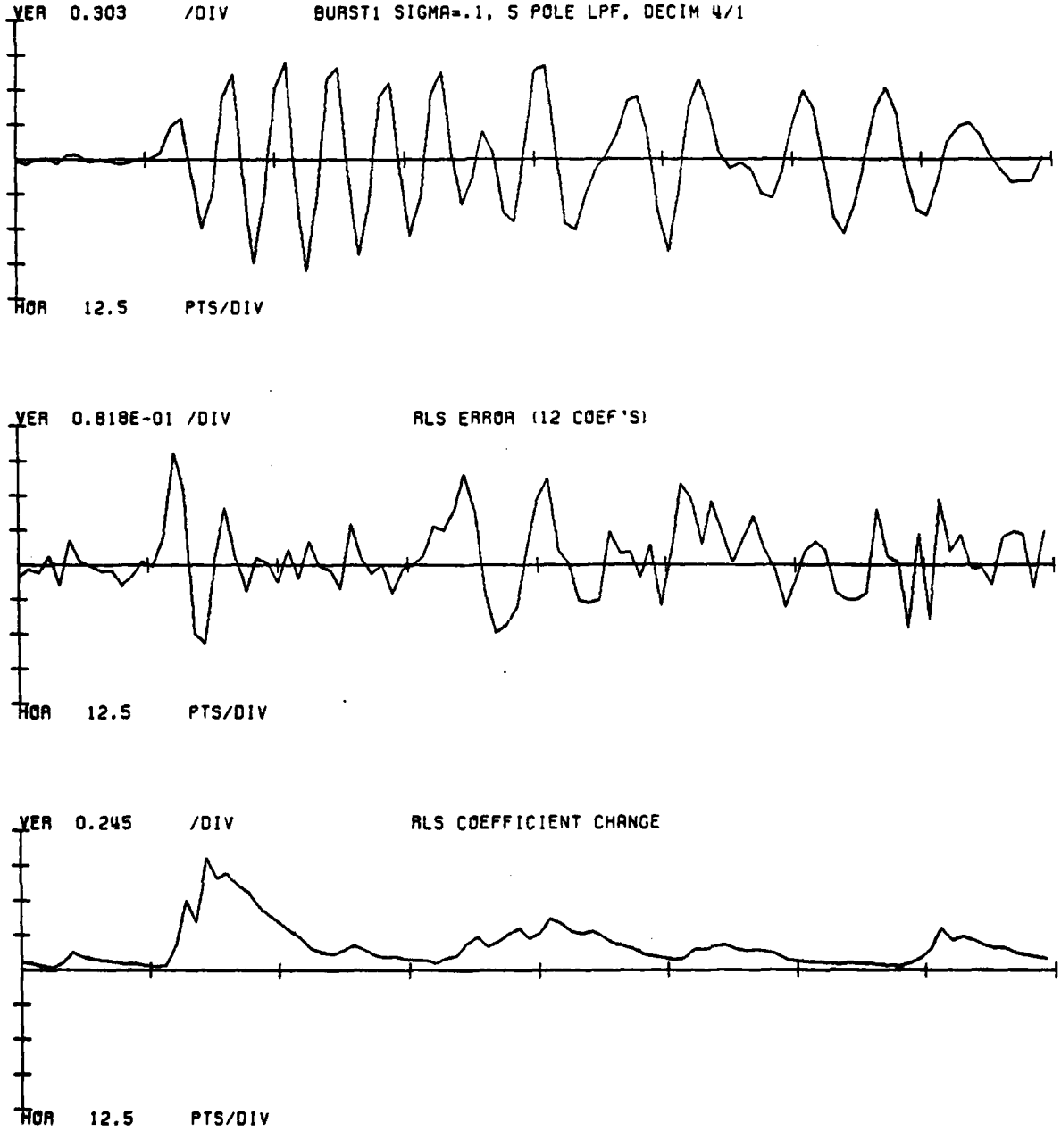

VER 0.303 /DIV
BURST1 SIGMA=.1, 5 POLE LPF, DECIM 4/1
HOR 12.5 PTS/DIV
VER 0.818E-01 /DIV
RLS ERROR (12 COEF'S)
HOR 12.5 PTS/DIV
VER 0.245 /DIV
RLS COEFFICIENT CHANGE
HOR 12.5 PTS/DIV

Figure 3.33c

3.6 Summary

The preceding examples indicate that both of these event compressed signals (the RLS error and the coefficient change signal) provide a means for locating the positions of events when those positions are not apparent in the input data. If the S/N ratio exceeds 40db, the compressed events are visible and despite the severe overlap of the input pulses, the compressed events are well separated. In addition, the duration of the compressed events is on the order of the predictor length and appears to be fairly independent of the existance or nature of previous events.

The following are some of the important results of these experiments.

> The starting location of the RLS iteration appears to only affect the sizes of the compressed events. However, there is a large initial pulse in the coefficient change signal due to the sensitivity of the algorithm to the first few data points. This means that the data must have several predictor lengths of noise preceding the first event if the coefficient change signal is to be used for event compression.
>
> Additive noise in the data does not have a substantial effect on the pulse shape of the compressed events. However, it does cause noise in the RLS error and coefficient change signals, which seems to be in proportion to the noise level in the data. Unfortunately, there is a large variation in the size of compressed events with this technique; making it difficult to establish a S/N criterion for event compression.

Compressed events do not influence each other substantially if the predictor has time to settle between them, even though the bursts which cause them may be severely overlapped. Figures 3.23a-h showed this by changing the relative position of the second and third bursts in an input sequence. The resulting compressed events had constant shape so long as they did not overlap. Therefore, the faster the predictor settles, the closer events can be to one another and still be resolved. This also means that events containing large numbers of zeroes (as in the FIR filtering examples) will not compress well with this technique unless they are widely separated. Note that a burst containing a large number of zeroes does generate a compressed event, but the event has longer duration than it would if the zeroes were not present.

While filtering the data may be necessary to reduce aliasing, we found that it did not improve the quality of the event compressed signals. If filtering must be performed, all-pole filtering should be used since the zeroes introduced by FIR filtering tend to lengthen the compressed events.

We found that decimation does increase the responsiveness of the predictor to each event, but the relative event to noise ratios in the location signals do not improve. In addition, the predictor settling time becomes a larger fraction of the event spacing, thereby increasing the likelihood of interevent interference. Consequently, decimation should be avoided; and in fact over-sampled data is likely to provide higher quality compressed events.

4. CONCLUSIONS

In this thesis we have developed an event compression technique based on the RLS algorithm, which can effectively compress events of differing spectra; provided they fit the model assumed by the RLS method (i.e. all-pole events). We examined the RLS prediction error as an event compressed signal and in addition developed the coefficient change signal for event compression. Our experiments indicate that this technique is effective only if the S/N ratio is high (e.g. > 40db) , but that variations in the event ordering, event positions or starting position of the algorithm have only minor effects on the compressed events. Therefore, this technique should be useful in those situations where the events are close to all-pole and the noise level is not high.

This thesis was an initial investigation into the feasability of using the RLS algoritm for locating events with differing spectra. When we began this work we wanted to discover if the technique had any value and if it did, where should further work be done. Given that need, the best course seemed to be to use the algorithm on a large number of synthesized examples, where we knew the number and positions of the events in the data. These examples indicate that RLS is indeed a viable means for performing event compression. However, there are many questions touched upon in the course

of this thesis which need more detailed investigation.

One of the most important results of this thesis is that the events in the data cannot be spaced closer than the predictor settling time, if they are to be resolved from each other. The issue of what determines this settling time remains unsolved. Our experiments indicate that it increases with noise level, decreases with predictor length (up to a point), and figures 3.23a-h indicate that it is not strongly dependent on the position of previous events. Also, the settling time for a given event is strongly dependent on the "character" of that event. More investigation of the dependence of the predictor settling on the event characteristics would be useful; insofar as it provides a means for decreasing the settling time of the predictor, thereby reducing the necessary separation between events.

Another possible avenue of investigation is that of alternatives to the RLS algorithm as presented in this thesis. Some variations of this adaptive algorithm exist which use a moving region Q over which the total squared energy E is minimized, rather than an expanding region as was done here. Still other methods involve exponentially weighting the past data. These modifications would allow the algorithm to be used over large amounts of data without the predictor becoming totally insensitive to the new points (as the error region

grows the predictor tends to become less sensitive). In some applications that capability might be essential; more importantly, reducing the amount of previous data to be predicted might allow the algorithm to more readily adapt to new events, thereby reducing the settling time. Our own feeling is that a means for dynamically changing the region of error minimization might be more effective than simply expanding or moving the region at each iteration, since that would permit the region of error minimization to be reduced at a new event to shorten the settling time, and then lengthened between events to reduce noise.

We found the coefficient change signal to be useful for event compression, but our signal used equal weights for all the coefficients and a fixed decay time for the filters. Is there an optimum choice for the weights of the coefficients in the change signal? Could Kalman filtering be used to track the coefficients more effectively? Perhaps there is a better alternative than simply high passing the coefficients. For example an adaptive decay time for the coefficient change signal might make the compressed events for the second burst in the Burst1 signal more visible.

Finally, there is no substitute for experiments on real data. The original motivation for this work was to find a means for compressing sonic well log data. Unfortunately,

subsequent study of the well logging problem revealed that the structure of the signals recorded in that situation does not fit the model that was assumed for this work. Experimentally we found that this technique did not compress those signals, but given their complicated structure, that was not surprising. Therefore, the application of this method of event compression to data which more closely corresponds to the assumed model is needed. We hope to perform experiments using acoustic cardiac data in the near future.

References

Deczky, A.G. (1972). "Synthesis of Recursive Digital Filters Using the Minimum-p Error Criterion", IEEE Trans. on Audio and Electroacoustics, vol. AU-20, no. 4, pp. 257-263

Eykhoff,P. (1974). SYSTEM IDENTIFICATION: Parameter and State Estimation, John Wiley and Sons, New York, p. 235

Makhoul, J. (1975). "Linear Prediction: A Tutorial Review", Proc. IEEE, vol. 63, no. 4, pp. 561-580

Markel, J. (1972). "The SIFT Algorithm for Fundamental Frequency Estimation",IEEE Trans. on Audio and Electroacoustics, vol. AU-20, no. 5, pp. 367-377

McClellan, J., Parks, T., Rabiner, L. (1973). "A Computer Program for Designing Optimim FIR Linear Phase Digital Filters", IEEE Trans. on Audio and Electroacoustics, vol. AU-21, no. 6, pp. 506-526

Satorious, E.H. (1979). "On the Application of Recursive Least Squares Methods to Adaptive Processing", Intl Workshop on Appl of Adaptive Control, Yale University

Tribolet, J. (1977). "Seismic Applications of Homomorphic Signal Processing", PhD Thesis M.I.T.

Ulrych, T. (1971). "Applications of Homomorphic Deconvolution to Seismology", Geophysics, vol. 36, no. 4, pp. 650-660

Young, T.Y. (1965). "Epoch Detection-A Method for Resolving Overlapping Signals",Bell Sys. Tech. Journal, vol. 44, pp. 401-426

Distribution List

Contract N00014-75-C-0951

	No. copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2
Office of Naval Research Branch Office/Boston Bldg. 114, Sec. D 666 Summer Street Boston, Massachusetts 02210	1
Office of Naval Research Branch Office/Chicago 536 South Clark Street Chicago, Illinois 60605	1
Office of Naval Research Branch Office/Pasadena 1030 East Green Street Pasadena, California 91106	1
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6
Commandant of the Marine Corps (Code AX) Dr. A. L. Slafkosky Scientific Advisor Washington, D.C. 20380	1
Office of Naval Research Code 455 Arlington, Virginia 22217	1
Office of Naval Research Code 458 Arlington, Virginia 22217	1
Naval Electronics Laboratory Center Advanced Software Technology Division Code 5200 San Diego, California 92152	1

Mr. G. H. Gleissner 1
Naval Ship Research and Development Center
Computation and Mathematics Department
Bethesda, Maryland 20034

Captain Grace M. Hopper 1
Office of Chief of Naval Operations
NAICOM/MIS Planning Branch
NOP-916D Pentagon
Washington, D.C. 20350

Advanced Research Projects Agency 1
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Commander 1
U.S. Army Electronics Command
Attn: Mr. J. L. DeClerk
AMSEL-NLY
Fort Monmouth, New Jersey 07703

Dr. Robert Kahn 1
Program Manager
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Director, National Security Agency 1
Attn: R54, Mr. Page
Fort G. G. Meade
Maryland 20755

ONR New York Area Office 1
715 Broadway, 5th Floor
New York, New York 10003